

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Editor terénu s využitím frameworku
XNA
Terrain Editor Based on XNA
Framework

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Martin Vlček**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Editor terénu s využitím frameworku XNA
Terrain Editor Based on XNA Framework

Zásady pro vypracování:

Cílem práce je navrhnout a implementovat editor terénů. Aplikace by měla být modulární a rozšiřitelná o možnost nahrávání vlastních modelů, renderování videí, apod.

Hlavní body zadání:

1. Přehled existujících nástrojů pro generování a editaci terénů.
2. Návrh a implementace editoru terénů ve zvoleném prostředí.
3. Vytvoření demonstrační scény.
4. Výkonnostní testy a jejich vyhodnocení.
5. Shrnutí dosažených výsledků.

Seznam doporučené odborné literatury:

Edward Angel: Interactive Computer Graphics, ISBN-10:032153586 (2009)

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Petr Gajdoš, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry





prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.



Podpis autora

Abstrakt:

Práce je zaměřena na počítačovou 3D editaci a tvorbu terénu. Obsahuje analýzu současných editorů a srovnání jejích funkcí. Dále se zabývá návrhem a implementací vlastního editoru. Grafická část editoru, určena k zobrazování terénu je implementována v prostředí frameworku XNA. Poslední část práce obsahuje výkonnostní testy tohoto editoru.

Klíčová slova:

editor terénu, .Net framework, XNA framework, level of detail(LOD), semínkové vyplňování

Abstract:

This thesis is focused on computer editing and creating 3D terrain. Includes analysis of the current editors and comparison of its functions. It also deals with design and implementation of its own editor. Graphs editor, intended for displaying terrain is implemented in an XNA framework. The last part contains performance tests of the editor.

Key words:

terrain editor, .Net framework, XNA framework, level of detail(LOD), seed filling

Obsah

1. Úvod	1
2. Cíle	2
3. Přehled existujících nástrojů pro generování a editaci terénů	3
3.1. TerraEd pro.....	3
3.2. Earth Sculptor.....	5
3.3. PnP Terrain Creator.....	7
3.4. Ostatní editory	8
3.4.1. T.ED	9
3.4.2. Grome.....	9
3.4.3. Terragen 2.....	10
4. Návrh implementace editoru terénů ve zvoleném prostředí	11
4.1. Popis zvoleného implementačního prostředí	11
4.2. Návrh implementace.....	11
4.3. Popis implementace.....	12
4.3.1. Použití XNA ve Windows Forms.....	12
4.3.2. Vykreslování terénu + LOD	12
4.3.3. Implementace vody	16
4.3.4. Logická struktura modelů.....	18
4.3.5. Komponenty třetích stran	19
5. Vytvoření demonstrační scény	22
5.1. Modelování terénu.....	22
5.2. Kreslení textur	24
5.3. Vkládání modelů	25
5.4. Vložení vody a vytvoření průletu scénou	26
6. Výkonnostní testy a jejich vyhodnocení.....	27
6.1. Popis testování.....	27
6.2. Test velikosti terénu	28
6.3. Test textur.....	29
6.4. Testování modelů	30
6.5. Test vody	30
6.6. Testování prezentačního videa	31
6.7. Vyhodnocení testů.....	32

7.	Shrnutí dosažených výsledků	34
8.	Další vývoj editoru	35
8.1.	Import a export terénu	35
8.2.	Vykreslování terénu	35
8.3.	Kreslení textur	36
8.4.	Použití objektů.....	36
8.5.	Voda	36
8.6.	Průlety scénou	37
9.	Závěr.....	38
10.	Seznam zkratk a pojmů.....	39
11.	Seznam obrázků	40
12.	Seznam tabulek, grafů a zdrojových kódů	41
13.	Literatura	42

1. Úvod

Editor terénu se dá využít v mnoha oblastech. Člověk si lépe udělá představu o výsledku ve 3D zobrazení, než v náčrtech nebo fotkách. I když máme náčrt nebo fotku z perspektivního pohledu nebo i klidně z více pohledů, nikdy nám to neposkytne potřebně detailní záběry, abychom pokryly všechny části zobrazovaného objektu nebo oblasti. V případě staveb si člověk nikdy nedokáže přímo představit, jak daná budova zapadne do okolního prostředí. Poté může nastat problém, kdy se původní návrh, který se zákazníkům líbil, může nakonec setkat se nesouhlasnými názory. 3D zobrazení terénu může také pomoci při modelování zahrad, parků, golfových hřišť nebo také dětských parků.

Další velkou oblastí pro využití editace terénu jsou počítačové hry. Zde je v mnoha případech potřeba vytvořit prostředí, kde se tato hra odehrává. Ať se již jedná o akční hry z prvního pohledu, kde většinou stačí zobrazit v určité chvíli vždy jen malou část terénu ovšem s mnoha detaily. Nebo strategické hry, kde je naopak potřeba zobrazit velmi rozsáhlé terény, ale s menším počtem detailů, neboť se kamera většinou pohybuje vysoko nad tímto terénem.

3D editory terénu mohou také sloužit k tvorbě virtuálních zobrazení existujících objektů. Například areálů škol, výstavišť a podobně. Zájemce pak může jenom zhlédnout na internetu video, které bude mnohem popisnější, než obyčejná mapa, kterou většina lidí zapomene. Respektive může být výsledkem interaktivní 3D mapa, na které se uživatel podívá, jak se má kam dostat nebo kde najde to, co potřebuje.

2. Cíle

Cílem mé diplomové práce je vytvořit editor terénu, který bude použitelný pro jednoduchou tvorbu terénů pro ukázky nebo pro virtuální prohlídky. Výsledkem by měla být možnost prohlížet si terén buď přímo v tomto programu ve 3D, nebo by mělo být možno vygenerovat video s průletem nad terénem. Kde si člověk zvolí dráhu letu kamery a program sám video vygeneruje. Jedním z požadavků by měla být jednoduchost ovládání. Aby se uživatel nemusel zatěžovat učením složitých menu a klávesových zkratk. Editor musí poskytovat možnosti k editaci výškové mapy. Především volbu velikosti této mapy a kreslicí štětce k editaci výšky. Dále musí poskytovat možnost nanášet více různých textur na mapu. Ty by se měli opět nanášet štětcem. Mělo by být umožněno volit si místa, kde bude zobrazována vodní hladina. A to nezávisle na sobě, aby vodní hladiny na různých místech mohli mít rozdílnou výšku hladiny. Vodní hladina by měla vypadat co nejvíce realisticky. Další možností je přidávat do scény objekty jako budovy, stromy a jiné objekty reálného světa. Mít možnost efektivně spravovat tyto objekty a to tak, že je bude možné seskupovat ve složkách a konfigurovat jejich vlastnosti buď jednotlivě, nebo pro celou složku. Samozřejmostí je možnost ukládat rozdělaný terén a poté ho zpět načítat. A posledním funkcí jsou dříve zmíněné průlety kamerou, ze kterých, by mělo jít vytvořit prezentační video.

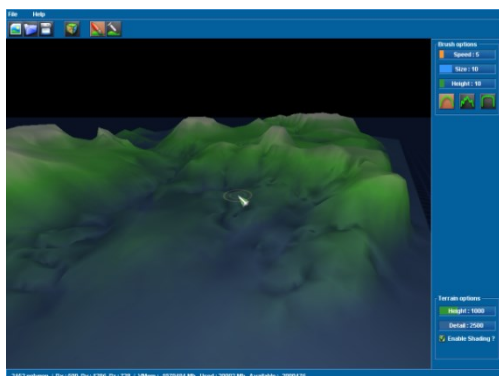
3. Přehled existujících nástrojů pro generování a editaci terénů

V současné době existuje několik editorů terénu. Tyto editory se v mnoha věcech odlišují a podle toho se dá určit, k čemu jsou primárně určené a co se na nich dá nejlépe dělat. Některé z nich jsou publikovány zadarmo, ale většinou to jsou komerční aplikace. Podle ceny lze také docela velmi dobře odhadnout kvalitu daného editoru. Co však mají společné je většinou velmi obsáhlé ovládání, kde člověk není sám schopný přijít na všechny funkce, respektive někdy je velmi složité nějakou funkci najít nebo pochopit jak jí použít. Tím, že se každý editor zaměřuje na něco jiného, vyplývá také problematika, kdy žádný samozřejmě neobsahuje všechny funkce. Některým chybí možnost přidávat objekty, což snižuje možnost vytvářet reálné terény.

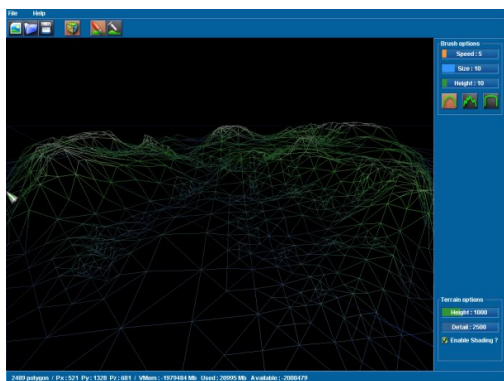
3.1. TerraEd pro

TerraEd patří mezi méně kvalitní editory a také výstup z něj nevypadá moc realisticky. Ale i přesto má několik velmi dobrých funkcí. Trochu nevýhodou mi přijde, že program je rozdělen na dvě oddělené části, na hlavní program pro správu scény a na část, kde se upravuje výšková mapa samotného terénu. Což může být trochu nepraktické, pokud si prve vytvoříme terén a pak nastavujeme celou scénu a zjistíme, že potřebujeme trochu někde terén upravit. Takže terén musí být připraven předem, respektive musíme přepínat program pokaždé, když chceme něco změnit ve výškové mapě. Tento editor je samozřejmě placený, ale je možné si stáhnout trial verzi pro vyzkoušení. Tato verze je omezená tím, že není možné nic ukládat. K této trial verzi je také přidána jedna hotová ukázková scéna. Placené verze jsou dvě, levnější, která obsahuje pouze editor. Druhá dražší má navíc také balíček obsahující demo mapy a 3D objekty.

První část pro úpravu výškové mapy nepatří mezi nejlepší. Ovládání je sice jednoduché a docela intuitivní, ale nedostatkem je malý počet štětců. Ty jsou jenom tři a každý se dá sice různě nastavovat, ale práce s nimi není již tak příjemná a jednoduchá. Mapu je možné si zobrazit jako drátěný model, pak ale štětec není moc vidět a pracuje se s ním ještě hůř. Ovládání je omezeno pouze na nastavování výšky v mapě anebo uložení výšky v určitém bodě na mapě do schránky. Terén je možné načíst z výškové mapy nebo vytvořit nový, zde mi ale chyběla možnost volit si velikost tohoto terénu. To se ale řeší tak, že pokud chci větší terén tak si jich vyrobím více a ty pak vložím v druhé části programu do scény a pokusím se je přizpůsobit k sobě. U drátěného modelu je vidět, že terén je na počátku tvořen dvěma trojúhelníky, které tvoří celý čtverec mapy a ty se pak při kreslení kopců přidáváním bodů zjemňují, tak že se rozpadají na menší trojúhelníky. Dále je poznat, že je použito technologie LOD, která zobrazuje bližší části terénu detailněji. Ukázky této editace jsou vidět na obrázcích 1 a 2.



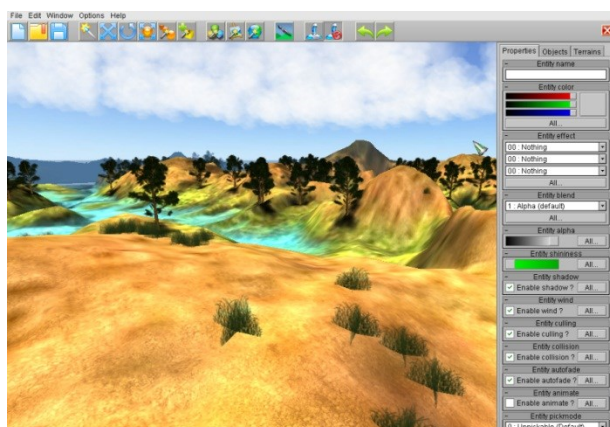
Obrázek 1: TerraEd editace terénu



Obrázek 2: TerraEd zobrazení drátěného modelu

Druhá část slouží ke kompletní správě scény, její ukázkou lze vidět na obrázku 3. Prostředí editoru nepatří mezi nejnovější a vypadá jako v mnohem starších programech. Ale ovládání je opět jednoduché, v editoru nejsou žádné složité menu a všechno je přehledné. Zde do scény můžeme vkládat jednotlivé terény, ty je možné vedle sebe skládat jak je potřeba. To ovšem nemusí být vždy ideální. Při pohybu nad terénem je velmi hodně vidět jak se přepočítává LOD a to nevypadá nejlépe. Co vidím jako výhodu je, že se nelze dostat kamerou pod terén, tedy pokud se pohybujeme nad terénem, mimo terén to lze. Co mi již moc praktické nepřišlo je nastavování textur na terén, vykreslování stínů a zobrazování detailů na terénu. Ty se do editoru vkládají jako textury přes celý terén, takže si je nemůžeme v tomto editoru upravovat, ale musíme je mít připraveny. Detaily nám určuje textura detailů, ta se ovšem dá vložit jenom jedna a používá se opakovaně pro celou mapu bez ohledu jaká část terénu to je. Program dále obsahuje generátor terénu, textur a mapy stínů, ten jsem bohužel otestovat nemohl, protože funguje také jenom v placené verzi. V dalším menu se dá měnit nastavení mlhy a je zde také nastavení vody, ale to bylo v mé verzi prázdné, takže nemohu říct, jestli se jedná o omezení trial verze nebo v programu implementováno není. V programu nalezneme nastavení skyboxu, kde si můžeme pro každou stranu volit obrázek, nastavení slunce, tomu se dá nastavit pozice a taky textura. A také se dá nastavit textura mraků, kdy se nastavuje jejich počet,

velikosti a vítr, podle kterého se tyto mraky budou pohybovat. Za velmi dobré považuju zpracování správy modelů. Importovat modely je možné ve více formátech a to x file, 3ds a B3d. Modely pak můžeme buď vkládat po jednom do scény kliknutím na místo, kde ho chceme, duplikováním z objektu, který již ve scéně je, nebo můžeme použít náhodné vkládání objektů do scény. Kde si zvolíme, kolik jich chceme přidat, rozptyl velikosti a rotaci a modely se pak ve scéně náhodně rozmístí.



Obrázek 3: TerraEd správa scény

3.2. Earth Sculptor

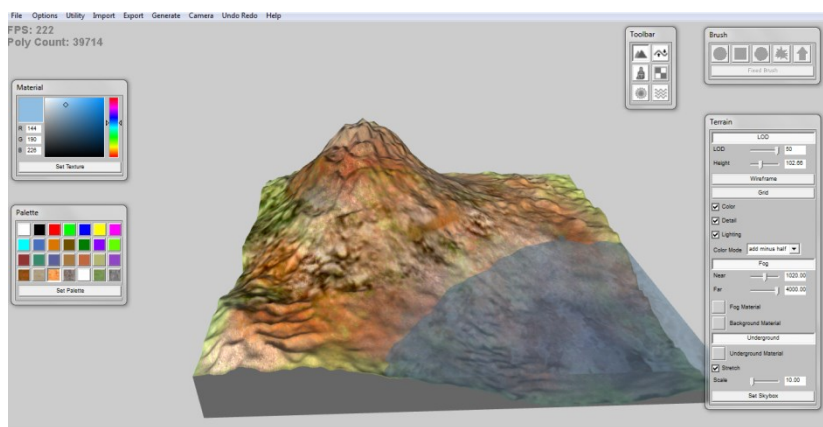
Earth Sculptor patří mezi velmi kvalitní editory s velmi dobře vypadajícími výsledky, ukázkou prostředí s vytvořeným terénem lze vidět na obrázku 4. U tohoto editoru bych chtěl hlavně pochválit ovládání, to je naprosto jednoduché a intuitivní. Dá se během chvíle přijít a pochopit všechny funkce tohoto editoru. V ovládání je tento editor podle mě nejlepší ze všech, které tady popisuji. Tato aplikace je opět placená, a je možné stáhnout si trial verzi na vyzkoušení. V trial verzi je možné používat všechny možnosti této aplikace s několika omezeními.

	Trial	Plná verze
Počet štětců	5	40
Maximální velikost mapy	257 x 257	4097 x 4097
LOD optimalizace exportovaného objektu	Ne	Ano
Komerční využití	Ne	Ano

Tabulka 1: Porovnání verzí editoru Earth Sculptor

Terén se kromě vlastní editace může naimportovat z externí výškové mapy, nebo je možno použít generátor výškové mapy. Aplikace má také generátor pro kreslení textur. V editoru je samozřejmě použito LOD, které je ovšem možno volitelně vypnout. Terén se dá zobrazit v režimu drátěného modelu, ten mi ovšem nepřipadá velmi povedený, protože tloušťka čar je velmi hrubá, takže drátěný model není moc dobře vidět a jednotlivé čáry se ztrácejí v rovnoměrné ploše. Dále je možné si zobrazit mřížku na mapě pro jednodušší modelování. Velmi dobře je zpracováno modelování terénu. Během chvilky se dá namodelovat libovolný terén i při menším počtu štětců v trial verzi. Editor totiž poskytuje pro každý štětec 8 způsobů jak tento terén modelovat, u každého se dá samozřejmě nastavovat velikost, intenzita a jemnost na okrajích. Kreslení textur se dělí na dvě části. V první části je možné navolit si 4 základní textury a ty pak nanášet na terén, tyto textury lze načíst z externích souborů. Možnosti kreslení terénu rozšiřuje obarvování štětcem 6 dalšími barvami, ty si můžeme navolit s předpřipravených barev, nebo si vytvořit kombinací načtené textury a barvy. Tento přístup nám dává při kreslení spoustu možností, jak může terén vypadat. Voda je zpracovaná jako jedna velká rovina přes celý terén, takže když si nastavíme výšku vodní hladiny a všude, kde se dostaneme terénem pod tuto úroveň, budeme mít vodu. Dále tam je mnoho různých nastavení této vody, které přidávají na reálnosti. Samozřejmostí je také nastavení osvětlení, skyboxu nebo mlhy.

Jako výstup z editoru je možné vytvářet statické obrazy přímo při tvorbě, nebo je možno vygenerovat terén jako model ve formátu obj. To je velmi výhodné v případě dalšího použití terénu v dalších aplikacích. Editor je primárně určen pro tento výstup, takže tolik nevádí, že nelze přidávat do scény objekty, protože terén stejně použijeme v jiné aplikaci, kde tyto objekty přidávat půjde, a budou se vkládat na tento terén. I když v některých případech je výhodnější vkládat tyto objekty, již při tvorbě samotného terénu, scénu pak můžeme mít jako celek i s objekty.



Obrázek 4: Ukázka editoru Earth Sculptor

3.3. PnP Terrain Creator

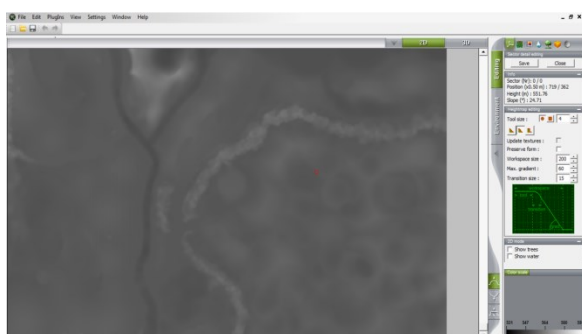
Tento editor je sice z jedné strany velmi zdařilý ale zase naopak obsahuje mnoho nelogických věcí, které by mohly být udělány lépe nebo jinak. Celkově se ale funkčností a možnostmi řadí na první místo z předchozích testovaných editorů. Tento editor lze získat ve 4 verzích, základní verze je zadarmo a další vždy zvyšují funkčnost, možnosti a v neposlední řadě také cenu. Další výhodou je, že lze z internetu stáhnout rozšiřující balíky textur, modelů, zvuků a také 2 ukázkové scény. Další výhodou oproti předchozím editorům je, že v instalačním balíku je přidán pdf soubor s kompletním manuálem k editoru, v kterém jsou vysvětleny všechny funkce aplikace.

Editor má několik vlastností, které většina ostatních postrádá. První touto vlastností je možnost vytvářet velmi velké terény, tato vlastnost je samozřejmě omezena podle verze. Možnost spravovat tyto velké terény, je dána tím, že terén je rozdělen na několik čtvercových sektorů a tyto sektory se upravují samostatně. U nejdražší verze je tedy možno mít až 999 x 999 sektorů. Další zajímavostí je možnost dívat se na terén v 3D pohledu nebo 2D, takže terén vidíme přímo kolmo shora a můžeme ho v tomto pohledu také upravovat, tyto pohledy jsou vidět na obrázcích 5 a 6. Prvním slabším místem editoru je pohyb kamerou. Otáčení a přibližování funguje bez problému, ale posun do stran je buď velmi pomalý anebo s klávesou shift zase velmi rychlý.

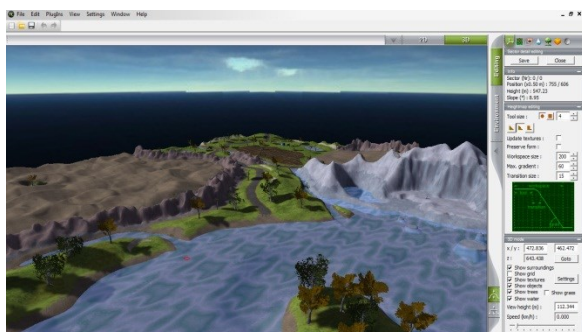
Samotné upravování terénu považuju za nejslabší část tohoto editoru. Editor má dva módy v prvním lze zvedat terénem nahoru a dolů a v druhém jsou štětce na vyhlazování terénu. Nastavení tvaru štětce lze jenom na kruh nebo čtverec, dále lze měnit velikost a intenzitu a tvar okrajů. Předchozí editor již v bezplatné verzi poskytoval mnohem více možností. Texturování má zase výhody i nevýhody. Pokud vytváříme nový terén a klikneme na kreslení textur, vyhodí nám aplikace chybovou hlášku. Ty se totiž musí nejprve nahrát do složky TEXTURES, která je v místě, kde máme terén uložený. Z programátorského hlediska to jistě bylo výhodné, protože není potřeba tyto textury již nijak ukládat. Pro uživatele mi ale přijde lepší načíst si ty textury z nějakého globálního umístění a pak je automaticky uložit, než nutit kopírovat uživatele textury do složky s terénem ručně. Naopak výhodou je to, že lze mít těchto textur neomezeně, jejich počet bývá u většiny editorů omezen jako například u předchozího editoru, kde mohly být maximálně 4. Kreslení textur probíhá úplně stejně, jako u úprav terénu. K dispozici máme dva tvary štětce a navíc můžeme ještě nastavovat intenzitu krytí. Další zajímavostí je, že lze nabrat již nakreslenou texturu z terénu kliknutím myši na pozici v mapě. Textury je také možné nanést na terén náhodně, kde je možné volit, kolik procent bude která textura zabírat.

U vkládání vody, se klasicky nastaví výška vodní hladiny a terén se proloží rovinou s texturou vody. Tato voda je průhledná ale nepohyblivá a bez odlesků, takže nebyl použit žádný shader. A to kazí výsledný realistický efekt. U objektů se opět chová stejně jako u textur, je potřeba mít objekty nahrané ve správných složkách, do kterých je potřeba nakopírovat ručně. Tyto objekty jsou děleny na rostliny a ostatní objekty. Přičemž rostliny lze přidávat hromadně pomocí generátoru náhodného rozmístění. Velmi dobré je zpracování stínů, zde je možné si nastavit kromě místa, odkud svítí světlo, také i barvy stínů, barvu světla a jestli se bude jednat o svit slunce nebo měsíce.

Další silnou stránkou tohoto programu je exportování hotového výsledku. Program umožňuje exportování celé scény do formátů, které používá samotný editor. Ale také dovoluje exportovat scénu po částech nebo jenom některé části a to do mnoha různých formátů. Je možné exportovat výškovou mapu do obrázku nebo RAW souboru. Umožňuje exportovat mapu textur a mapu stínů do nativního formátu pro editor nebo také do obrázků. Rozmístění objektů, porostu a zvuků tak, aby je bylo možno otevřít v editoru. A nakonec také výškovou mapu exportovat jako objekt do formátu 3DS, B3D, Obj, OGRE, a jako X soubor. Při používání aplikace jsem také objevil několik problému, kdy aplikace občas padala. Většinou se to stalo při připojení externího monitoru nebo při uspání počítače.



Obrázek 5: PnP 2D zobrazení terénu



Obrázek 6: PnP 3D zobrazení ukázkové scény

3.4. Ostatní editory

Dále bych chtěl ještě zmínit několik dalších editorů. Vybral jsem několik, které patří mezi kvalitní editory a stojí tedy za zmínku. Kvalitou jsou srovnatelné s dříve popisovanými editory a některé z nich je kvalitou dokonce i převyšují, to vše je ale na úkor jednoduchosti ovládání, které je mnohonásobně obtížnější. K většině editorů lze ovšem najít na stránkách výrobce video návody s popisem ovládání.

3.4.1.T.ED

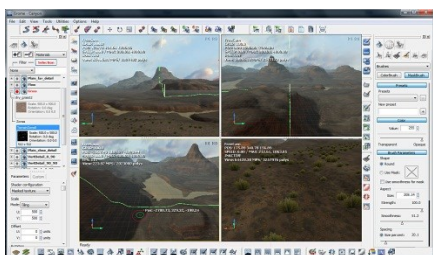
Tento editor bych kvalitou srovnal s editorem TerraEd. Výsledné scény vypadají sice docela dobře, dokonce mnohem lépe než v TerraEdu. Ovšem trochu nepříjemné je ovládací prostředí, v tom zase mnohem zaostává. Tento editor byl poprvé vydán v roce 2005, poté byl upravován až do roku 2008. Ovšem prostředí vypadá mnohem starší. Ovládaní je řešeno úplně jinak, než člověk očekává a používá se v ostatních editorech a také samotné ovládací prvky nejsou moc dobře čitelné. Ikonky v editoru mají obrázky, ze kterých není poznat co na nich je a také popis je malý a téměř nečitelný. Při modelování terénu editor přeblikává z zobrazení textur na zobrazení s mřížkou, to působí již po chvilce práce velmi nepříjemně na oči. Výsledky z tohoto editoru nevypadají špatně, ale ovládaní způsobuje, že je tento editor velmi podprůměrný. Editor je naštěstí nabízen za poměrně nízkou cenu oproti ostatním, což může vykupovat jeho nedostatky.



Obrázek 7: T.ED

3.4.2.Grome

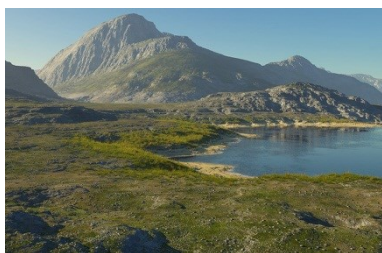
Grome je velmi kvalitní editor, který využívají také některé větší vývojářské studia počítačových her. To samo o sobě dokazuje jeho kvality. Výsledky tohoto editoru jsou velmi realistické a autorům se povedlo velmi pěkně zpracovat mnoho funkcí, které tento editor poskytuje. Jako jedinou vadu bych viděl zpracování některých ovládacích prvků. Autoři moc nehleděli na jednoduchost a intuitivnost ovládaní a prostě většinou tlačítek doslova vyplnili všechna volná místa v editoru. To dělá tento editor trochu nepřehledný. Autoři naštěstí na svých internetových stránkách poskytují video návody. Pomocí kterých se toto ovládaní dá naučit. Tento editor je již prodáván jako třetí verze, s tím že první dvě jsou stále v prodeji za nižší cenu.



Obrázek 8: Grome ovládací prostředí

3.4.3. Terragen 2

Tento editor patří mezi nejlepší editory terénu. Poskytuje fotorealistické výsledky. Také ho proto používají filmová studia pro vytváření videí s průlety terénem. Velký počet funkcí a vlastností tohoto editoru je ovšem vykoupeno mnohem obtížnějším ovládáním a v neposlední řadě také mnohem vyšší cenou, než ostatní editory. Je poskytován v několika verzích, podle ceny. Dražší verze obsahují samozřejmě více funkcí a také kolekce objektů. Nejdražší verze je téměř dvakrát dražší než Grome.



Obrázek 9: Terragen 2

4. Návrh implementace editoru terénů ve zvoleném prostředí

V této kapitole se budu věnovat vlastnímu návrhu a popisu samotné implementace editoru terénu. Návrh aplikace a implementace probíhaly podle požadavků, které byly popsány v kapitole [Cíle](#). Navrhovaná aplikace by měla běžet pod operačním systémem Microsoft Windows. Pokud bude potřeba, tak bude pro aplikaci vytvořen instalační soubor, který nainstaluje samotnou aplikaci a také všechny komponenty potřebné pro běh.

4.1. Popis zvoleného implementačního prostředí

Pro zobrazování terénu jsem zvolil Microsoft XNA. XNA je platforma, který slouží k tvorbě 3D a 2D grafických programů, které běží v Microsoft Windows, Windows Phone a Xbox 360. Vývoj probíhá v jazycích Visual Basic a C#. Já jsem si pro aplikaci zvolil jazyk C#. Microsoft XNA umožňuje vykreslování 2D/3D grafiky. Dále umožňuje přímo přistupovat k ovládacím prvkům jako je myš a klávesnice a zpracovávat data z těchto prvků. XNA umožňuje přistupovat přímo ke grafické kartě a využívat její výhody při vykreslování 3D grafiky. Práce s grafikou je tak usnadněna pomocí použití vertex bufferů, transformačních matic a jednoduchému přístupu k vykreslování modelů.

Ovládací prostředí se dalo napsat také v XNA, kde by byly všechny ovládací prvky grafické, ale to by bylo velmi náročné a zdlouhavé. Zvolil sem tedy knihovnu Windows Forms frameworku .Net. Tato knihovna umožňuje jednoduchou tvorbu uživatelského prostředí. Tato knihovna poskytuje spoustu předpřipravených ovládacích prvků, které stačí nastavit v grafickém rozhraní. Dále je možné tyto prvky skládat do celých komponent a ty používat, nebo je možné najít na internetu spoustu již zpracovaných open source komponent. U kterých nalezneme úplnou dokumentaci a většinou ukázkové použití.

4.2. Návrh implementace

Aplikace se bude skládat ze dvou hlavních částí. Hlavních formulář, který bude poskytovat uživateli ovládání aplikace a XNA komponenty, která bude vykreslovat terén na obrazovku. Důležité je vyřešit komunikaci mezi těmito dvěma prvky. Samotné zobrazení XNA ve Windows Forms budu přesněji popisovat později. Také je potřeba vyřešit aby se obě tyto hlavní části synchronizovali v nastavení, která funkce je zrovna používána. Dále je potřeba vyřešit ovládání kamery ať je nezávislé na ostatních funkcích editoru, to je potřeba aby se dalo pohybovat po terénu při používání jakékoliv funkce editoru. Jednotlivé ovládací prvky ve formuláři budou tvořeny jako kompletní komponenty, které budou obsahovat celou funkční logiku, potřebnou pro svou funkčnost. Pro některé části aplikace je možné použít již hotové komponenty třetích stran. Zde se bude jednat o

komponentu, která nahrazuje standardní TreeView ve Windows Forms a modul pro tvorbu videa. Nahrazení TreeView je potřeba z důvodu, že je jeho použití nedostačující. Standardní TreeView totiž dovoluje uložit jenom text a index. Já potřebuji uložit si do něj svou vlastní strukturu a také přidat do stromu ikony, které budou moct po kliknutí vyvolat potřebnou funkci.

4.3. Popis implementace

V této kapitole budu popisovat implementaci. Konkrétně budu popisovat vybrané složitější algoritmy, u kterých není řešení na první pohled jasné nebo je více možností jak tento problém vyřešit. Implementaci některých jednodušších algoritmů popíšu v kapitole: Tvorba demonstrační scény. To budou hlavně algoritmy, které slouží k samotné tvorbě terénu a výsledné scény, respektive k vytvoření reprezentačního videa.

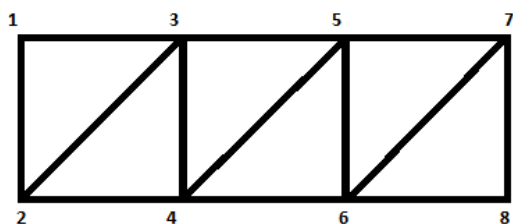
4.3.1. Použití XNA ve Windows Forms

Základním problémem pro tvorbu editoru byla kombinace uživatelského prostředí a vykreslovací komponenty. První možností bylo použít samotné XNA a vytvořit ovládací prostředí přímo v něm. Vytváření tlačítek a nastavování jejich chování by ovšem bylo velmi zdlouhavé a zbytečné, když Windows Forms poskytují jednoduchý přístup a také jednoduchý, ale účelný vzhled. Prvním problémem tedy bylo vytvoření komponenty, která obsahuje vykreslovací jádro XNA a je možné ji přidat do formuláře. K tomuto jsem použil již hotové řešení, které najdeme v zdrojích [\[1\]](#) a [\[2\]](#). Kombinací těchto řešení jsem aplikoval jako abstraktní třídu a z té dědím v hlavní komponentě pro vykreslování. Tato třída musí obsahovat základní prvky Graphics device a Content. Graphics device slouží k vykreslování objektů a nastavování grafických funkcí. Content slouží k načítání modelů, textur a efektů. Dále několik základních metod, nejdůležitější je metoda draw, ta slouží k samotnému vykreslování scény. A také obsahuje metodu pro inicializaci komponenty a metody k obsluhu akcí klávesnice a myši.

4.3.2. Vykreslování terénu + LOD

Vykreslování terénu je řešeno pomocí vertex bufferů, které obsahují pole vertexů. Dále je pro vykreslení potřeba pole indexů. Indexy jsou ve formátu triangle stripu, více najdeme v [\[3\]](#). Indexy se

zadávají v pořadí jako na obrázku 10. Pro vykreslení tohoto pásu stačí pouze 8 indexů, pokud bychom vykreslovali každý trojúhelník zvlášť, potřebovaly bychom 18 indexů. Jediným problémem této metody je, že je potřeba věnovat pozornost místům, kdy pás přechází na další řádek a zde je potřeba vykreslit trojúhelník navíc, který není vidět. Výhodou také je, že vykreslování pomocí vertex bufferů a triangle stripů podporuje přímo grafická karta a tím pádem je toto vykreslování mnohem rychlejší.



Obrázek 10: Triangle strip

I přesto je vykreslení velkého terénu velmi obtížné, u terénu 2000 krát 2000 vertexů, již vykresluje téměř 8 milionů trojúhelníků. A i samotné XNA má omezení při vykreslování na maximální počet 1 048 575 trojúhelníků při jednom vykreslení [4]. Možnosti tedy je, vykreslit celý terén v jednom cyklu na několikrát. Toto řešení je dobře použitelné, a pokud rozdělíme terén správně na pásy, pak nám stačí vytvořit jeden menší index buffer, který se použije pro každý pás. Problém je v tom, že budeme potřebovat zdvojovat vertexy na hranicích těchto pásu, aby mezi nimi nebyly mezery. A to nám ještě zvýší již tak velký počet trojúhelníků k vykreslení. Je tedy potřeba použít jinou technologii, která nám pomůže, abychom nemuseli vykreslovat všechny trojúhelníky. Pokud se díváme jenom na určitou část terénu a zbytek nevidíme, vyřeší tento problém za nás ořezové roviny. Pokud se ovšem díváme na celý terén je potřeba vykreslit všechny trojúhelníky.

4.3.2.1. LOD

To se dá vyřešit metodou LOD. LOD využívá principu, kdy nevykresluje všechny trojúhelníky, ale pouze ty, které potřebujeme. A to buď vykresluje více trojúhelníků v místě, které je blíže ke kameře, nebo vykresluje více trojúhelníků v místech, kde je terén členitější a tím pádem tam

potřebujeme větší detaily. Existuje spousta algoritmů, které se pro LOD používají. Tato kapitola čerpá z [5]

Algoritmy LOD se dělí na diskrétní a spojitý. U diskrétních se využívá předzpracování, kdy si nejprve připravíme několik verzí detailů a během vykreslování pouze vybíráme, kterou verzi pro vykreslení použijeme. Nevýhodou je větší paměťová náročnost, musíme mít v paměti uloženy všechny verze detailů, i když je třeba nepoužijeme. Ale výhodou je nezpomalování při běhu, zde se už nic nepočítá, jenom se vybere správná verze, kterou vykreslíme. Spojité LOD algoritmy generují trojúhelníkovou síť přímo za běhu aplikace. Tento přístup je tedy náročnější pro a ubere trochu výkonu aplikace, ale je možné dynamicky upravovat terén, tak aby bylo vidět jen to co je doopravdy potřeba.

4.3.2.1.1. ROAM

Mezi nejznámější spojitý LOD algoritmy patří ROAM. Základní implementací tohoto algoritmu je binární strom trojúhelníků. Ten se vypočítává rekurzivně dělením každého trojúhelníku na dva menší. Ty jsou pak potomky předchozího trojúhelníku. Nad binárním stromem se pak provádějí dvě operace spojení a rozdělení trojúhelníku. První snižuje počet detailů a druhá ho zvyšuje. Toto dělení a slučování se provádí vždy přímo při vykreslování. Nevýhodou tohoto algoritmu je, že je někdy potřeba provést vynucené dělení. Pokud jsme rozdělili trojúhelník, je potřeba také rozdělit jeho souseda, který s ním sdílí základnu. Po tomto rozdělení je potřeba zkontrolovat jestli nevznikl další trojúhelník, který má souseda se stejnou základnou a ten není rozdělený. To se řeší rekurzivním voláním.

4.3.2.1.2. Geo Mip-Mapping

Diskrétní algoritmus, který rozdělí terén na několik pravidelných dlaždic uložených v paměti s různou úrovní detailů. Dělení trojúhelníků je zde možné vyřešit několika různými způsoby. Jednou možností je dělit dlaždice pomocí čtverců se stejnohlými uhlopříčkami, další možností je dělit trojúhelníky půlením nejdelší hrany. Tento algoritmus je poměrně jednoduchý, jedinou jeho nevýhodou je, že při různých úrovních sousedních dlaždic vznikají v terénu mezery a ty je potřeba vyplnit.

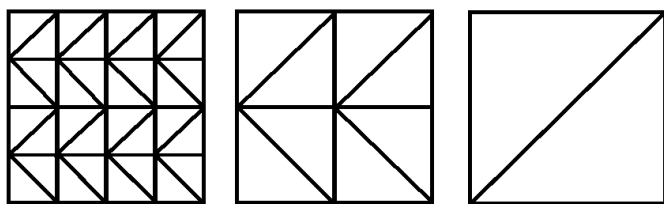
4.3.2.1.3. Chunked LOD

Tento algoritmus je velmi podobný předchozímu. Také využívá dlaždic a pevně předzpracovaných verzí detailů. Rozdíl je ovšem v tom, že tyto dlaždice nemusí mít stejnou velikost, ale zmenšují se s rostoucí úrovní detailů. Při nejnižší úrovni detailů pokrývá celá dlaždice celý terén. Při každé další úrovni se počet dlaždic zčtyřnásobí. Tak jako u předchozího algoritmu je potřeba vyplňovat mezery vzniklé mezi jednotlivými dlaždicemi v terénu.

4.3.2.1.4. Vlastní implementace

Z důvodu, že jsem se rozhodl přidat LOD do editoru až na konci celé implementace a to z důvodu navýšení výkonu aplikace. Hledal jsem řešení, které bude jednoduché a nebude již potřeba příliš měnit stávající funkčnost celého editoru. Z těchto důvodů jsem zvolil metodu Geo Mip-Mapping, která byla velmi jednoduchá na zapracování do téměř hotového editoru, a vykazovala dostačující výsledky ve výkonu. Tento algoritmus rozdělí terén nejprve na několik menších čtverců. Tyto čtverce jsou o velikost 257 x 257 vertexů, a to přidává omezení při tvorbě nového terénu na jeho velikost, ta musí být násobkem této velikosti jednoho čtverce. Pro rozdělení na několik úrovní detailů jsem použil vytvoření několika index bufferů. Každý index buffer, má rozdílnou hustotu vertexů. To mi umožňuje vykreslovat jednotlivé čtverce v různých úrovních detailů. Tyto index buffery si připravím na začátku běhu editoru. Výhodou je, že pro všechny čtverce mohu používat stejné index buffery. Nemusím je tedy vytvářet pro každý čtverec zvlášť. V editoru jsem vytvořil 5 úrovní detailů. Na začátku se tedy tvoří 5 index bufferů s různou hustotou vykreslení vertexů. Na obrázku 11 jsou vidět 3 úrovně vykreslování detailů. Před každým vykreslením je potřeba spočítat pro každý čtverec úroveň, kterou se má vykreslit. To je ovšem vykoupeno mnohem menším počtem trojúhelníků, které je potřeba vykreslit a to urychlí i celé vykreslování. Jednou nevýhodou tohoto jednoduchého algoritmu ovšem je to, že tvoří nespojitý terén a vznikají mezi čtverci s rozdílnou úrovní detailů díry. Zde existuje zase několik způsobů jak tento problém spravit. Já jsem použil algoritmus, který vytvoří mezi čtverci pásy. Tyto pásy vyplní vzniknuté mezery mezi jednotlivými čtverci.

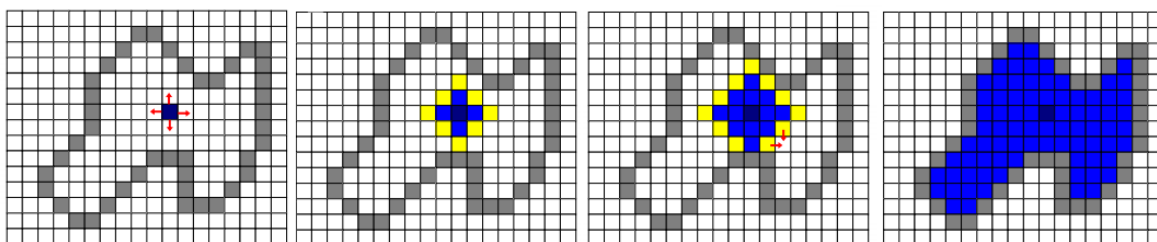
Velmi důležité je také ještě správně určit, jak spočítat úroveň každého čtverce. Zde jsem nejprve počítal vzdálenost čtverce od kamery, to se ale ukázalo jako nepraktické, protože se nejjemněji zobrazovali čtverce, které byly sice nejbliž kameře, ale méně detailů měly zase ty, na které se uživatel zrovna díval, neboť byly vzdálenější. Další logický nápad tedy byl určovat tuto úroveň podle toho, jak je čtverec vzdálen od místa, na které míří kamera. To ale také nebylo ideální, protože i při maximálním oddálení, byly tyto čtverce vykreslovány s maximálními detaily. Pak jsem tedy vybral pozici na, kterou míří kamera, ale vzal jsem ještě v úvahu oddálení kamery od terénu.



Obrázek 11: 3 úrovně Geo Mip-Mapping LOD

4.3.3. Implementace vody

Implementaci vody jsem se rozhodl udělat jiným způsobem, než jsme viděli v ostatních editorech. Ostatní proloží scénu rovinou vody a v místě, kde terén klesne pod tuto hodnotu je voda zobrazena. V cílech jsem si vytyčil předpoklady, že voda bude pouze v místech, které si vybereme a to i s rozdílnou výškou hladiny. Bylo tedy potřeba vymyslet princip, jakým bude uživatel určovat, kde má být voda v terénu a pak i samotný princip jakým se bude do terénu vkládat a vykreslovat. Základní myšlenka uživatelského přístupu je, že uživatel nejprve určí výšku vodní hladiny pomocí šipek, kde se mu bude zobrazovat voda pomocí jedné roviny přes celý terén. Vodní hladina tak bude vidět v místech, kde tato rovina bude nad terénem. Uživatel potom klikne myší, kde vodu opravdu chce a ona se potom přidá automaticky, respektive druhým tlačítkem myši půjde tuto vodu odebrat. Zobrazování roviny, která určuje výšku vodní hladiny přes celý terén je velmi jednoduché, ale mnohem obtížnější je samotné vyplnění vodní hladiny v místě, kde uživatel klikne myší. Nejprve získáme vertex, poblíž kterého uživatel klikl myší. Poté je potřeba zjistit jestli uživatel klikl na místo, které je pod vodní hladinou. A pokud ano tak rozšířit vodní hladinu v místech, která jsou pod touto hladinou. Zde jsem použil Algoritmus semínkového vyplňování, nalezneme zde [6]. To slouží k vyplňování nesouvislých spojitých oblastí. Algoritmus se používá pro rastrové vyplňování a je velmi jednoduchý. Většinou se používá jeho rekurzivní verze, kdy se volá funkce pro každý bod a ta vždy zkontroluje, jestli byl již daný bod přidán, pak jej přeskočí nebo patří mezi ty, které hledáme a pokud ano pak jej přidá do připraveného listu výsledných bodů. A v případě, že tento bod byl ten, který požadujeme, zavoláme tuto funkci i na okolní body.



Obrázek 12: Semínkové vyplňování

V editoru jsem každou část vody implementoval jako list trojúhelníku získaný právě touto metodou. Výhoda vkládat do listu přímo trojúhelníky než samostatné body je v tom, že pokud bychom vkládaly jenom body, tak by potom ještě bylo potřeba skládat z těchto bodů trojúhelníky. Pokud tam vkládáme přímo tyto trojúhelníky, stačí je potom už jenom vykreslit. Těmto trojúhelníkům stačí jenom nastavit výškovou souřadnici na hodnotu vodní hladiny, kterou si uživatel prve nastavil. Semínkové vyplňování jsem ale zvolil nerekurzivní verzi a to z důvodů, že u hodně velkých vodních ploch rekurzivní verze zabrala hodně paměti a někdy i celá aplikace spadla na paměť. Nerekurzivní verze se implementuje pomocí zásobníku, do kterého vkládáme jednotlivé trojúhelníky a poté je postupně ze zásobníku vybíráme a zpracováváme. Na počátku do zásobníku vložíme první trojúhelník. Algoritmus běží v nekonečné smyčce, dokud není zásobník prázdný, to nám zajistí, že vyplníme všechny trojúhelníky.

```
while (stack.Count != 0)
{
    index = stack.Pop();
    if (index < 1 || index > ((width - 1) * 2 * (height - 1 - heightCorrection) - 1) || water.UsedTriangle.Contains(index))
    {
        continue;
    }
    TerrainMapVertex[] triangle = map.getTriangle(index);
    for (int i = 0; i < 3; i++)
    {
        if ((triangle[i].Position.Y - water.SurfaceHeight) > 0)
        {
            count++;
        }
    }
    if (triangle[0].Position.Y < deepestPointValue)
    {
        water.DeepestPoint = index;
        deepestPointValue = triangle[0].Position.Y;
    }
    triangle[0].Position.Y = triangle[1].Position.Y = triangle[2].Position.Y = water.SurfaceHeight;
    List.Add(new VertexPositionTexture(triangle[0].Position, new Vector2(triangle[0].TextureCoordinate.X, triangle[0].TextureCoordinate.Z)));
    List.Add(new VertexPositionTexture(triangle[1].Position, new Vector2(triangle[1].TextureCoordinate.X, triangle[1].TextureCoordinate.Z)));
    List.Add(new VertexPositionTexture(triangle[2].Position, new Vector2(triangle[2].TextureCoordinate.X, triangle[2].TextureCoordinate.Z)));
    water.UsedTriangle.Add(index);
    if (count < 2)
    {
        if ((index + 1) % ((width - 1) * 2) != 1){
            stack.Push(index + 1);
        }
        if ((index - 1) % ((width - 1) * 2) != 0) {
            stack.Push(index - 1);
        }
        stack.Push(index - (width - 1) * 2);
        stack.Push(index + (width - 1) * 2);
    }
}
```

Zdrojový kód 1: Semínkové vyplňování

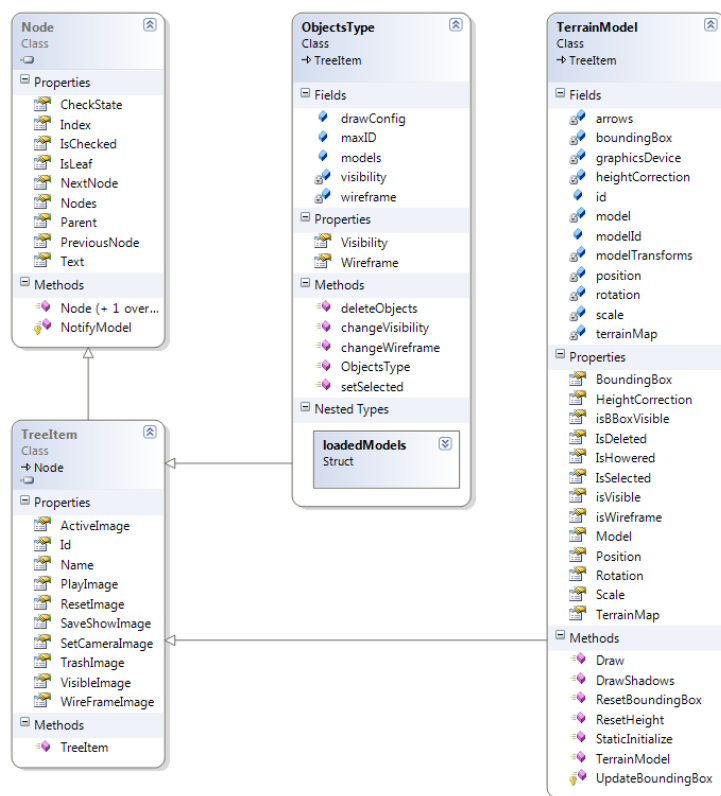
V zdrojovém kódu 1 lze vidět smyčku pro semínkové vyplňování. Metoda pracuje s celými trojúhelníky. Před touto smyčkou se vloží do zásobníku první trojúhelník, získaný kliknutím do terénu. Na začátku se vybere jeden trojúhelník ze zásobníku, u něj se zkontroluje, jestli neleží mimo náš terén, tyto trojúhelníky jsou brány pouze podle indexů, takže je jednoduché dostat se mimo tento terén. Dále se získají krajní body tohoto trojúhelníku a proběhne kontrola, jestli tento trojúhelník leží pod hladinou vody. Pokud ano přidám jej do připraveného listu. Dále pokud alespoň dva body leželi

pod hladinou vody, přidáme okolní trojúhelníky do zásobníku. Trojúhelník se přidává do listu s vodou pokaždé, i když by tam již neměl patřit a to z důvodu, aby na okrajích vodní hladiny nevznikaly díry. V zdrojovém kódu 1 je také vidět, se ukládá trojúhelník s největší hloubkou. To slouží pro jednoduchou kontrolu, kdy uživatel vloží vodu dvakrát na stejné místo. Nejhlubší trojúhelník je vždy stejný a podle toho můžu nalézt vodu, která je již na tom samém místě. Starou vodu poté smažu a nechám jenom nově vloženou.

4.3.4. Logická struktura modelů

Přidávat modely do scény je možné bohužel jen s připravené kolekce. Toto omezení bohužel způsobuje samotné XNA, protože to umožňuje pouze načítání modelů zkompileovaných Visual Studií a umístěných v Contentu. Content je kontejner, který obsahuje každý XNA projekt a do něj se umísťují všechny modely, obrázky nebo zvuky načítané v aplikaci. To ovšem určilo, že bylo potřeba již v samotné implementaci vytvořit kolekci modelů, které bude potřeba přidávat. Tyto modely jsem získal z internetových kolekcí bezplatných modelů.

Nyní bylo potřeba vyřešit jednoduchý přístup k ovládání těchto modelů. Tyto modely je potřeba posunovat, otáčet s nimi a měnit jejich velikost. A to někdy chceme i pro několik modelů najednou. Musela se tedy vymyslet logická struktura, která bude obsahovat stromovou strukturu složek a objektů. Tuto strukturu na víc bylo potřeba napojit na zobrazovací strom. Tento zobrazovací strom budu popisovat v kapitole [Advanced TreeView](#). Na obrázku 13 je vidět třídní diagram pro tuto strukturu. Vytvořil jsem tedy dvě třídy, které budou sloužit pro modely a složky do kterých se budou tyto modely ukládat. Obě tyto třídy dědí ze třídy `TreeItem`, ta slouží k zobrazování pomocí stromu a sama dědí ze třídy `Node`. Ta je součástí tohoto vykreslovacího stromu. Třída `ObjectType` je použita jako složka do které můžeme ukládat modely nebo další složky. To nám umožňuje vytvořit stromovou strukturu. `TerrainModel` slouží pro samotné modely a také obsahuje další prvky pro nastavení každého modelu. `TerrainModel` neobsahuje přímo tento model, ale pouze referenci na tento model. To je potřeba proto, aby se nemusel každý model načítat několikrát, což by zabíralo spoustu paměti. Každý model se tedy načte jenom jednou do statického listu v třídě `ObjectType`, tu se k němu také uloží jeho id pro tento model a pokud chceme tento model přidat do scény znova, podíváme se prve do tohoto listu a pokud zde již je uložíme si pouze referenci na tento model.



Obrázek 13: Logická struktura modelů třídní diagram

4.3.5. Komponenty třetích stran

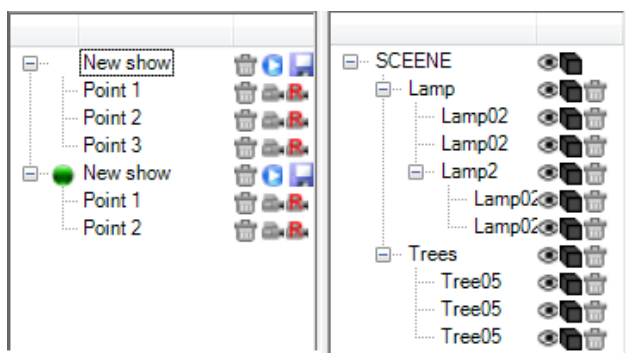
V editoru jsem použil také komponenty třetích stran. První komponentu jsem popsal již v kapitole [Použití XNA ve Windows Forms](#), kde se jednalo o vykreslovací komponentu pro XNA ve Windows Forms. Tato komponenta musela být v mnoha věcech upravena pro použití v tomto editoru. Následující dvě komponenty byly použity, tak jak jsou poskytovány a byly na nich provedeny minimální změny.

4.3.5.1. Advanced TreeView

Advanced TreeView slouží jako náhrada za obyčejný TreeView, který poskytuje Framework Microsoft .Net, jeho popis nalezneme zde [7]. Ten totiž poskytuje jenom základní možnosti, kdy k jednotlivým uzlům ve stromu umožňuje vkládat pouze text a index. Ten jsem potřeboval na použití k ovládání nastavení a zobrazení modelů. Pro jednoduché ovládání jsem potřeboval vkládat do jednotlivých uzlů přímo svoje prvky. A dále jsem potřeboval přidávat k uzlům kromě textu také

ikonky, na kterých se bude dát vyvolat funkce kliknutím myši. Advanced TreeView všechny tyto možnosti poskytuje. Všechny třídy, které se budou používat jako uzly v tomto stromu, musí dědit z třídy Node, implementované v tomto stromě. Pak jim stačí jenom přidat jako atributy obrázky a ty provázat s nastavením samotného stromu. Ty se potom zobrazí u těchto uzlů ve stromě. Další výhodou tohoto stromu je jednoduché použití přetahování uzlů stromu. Jediné co je potřeba doimplementovat jsou omezení pro samotné přetahování, kde bylo potřeba zajistit, které uzly se mohou přesouvat.

Tento strom měl nejprve sloužit pro zobrazování logické struktury modelů a k ovládání zobrazování těchto modelů. Ale nakonec pro jeho jednoduchost použití jsem se rozhodl ho použít také u vytváření průletu scénou. Zde je možné vytvářet několik těchto průletů a každému potom přidávat body průletu. Zde bylo opět možné použít několik ikon pro ovládání těchto bodů i samotných průletů. Na obrázku 14 jsou zobrazeny obě použití stromu. A to první pro použití tvorby průletu kamerou a v druhém pro ovládání modelů.



Obrázek 14: Advanced TreeView

4.3.5.2. AVI File

V .Net Frameworku není implementována žádná základní knihovna pro tvorbu videa. Použil jsem tedy komponentu AVI File, informace nalezneme zde [8]. Tato komponenta umožňuje jednoduchý přístup vytvoření videa. Nabízí vytvářet video pomocí vkládání bitmap pro každý snímek. Výhodou je, že komponenta dokáže nastavit parametry videa pomocí prvního snímku, není tedy potřeba je nastavovat ručně. Stačí tedy jenom nastavit počet snímků za sekundu. Dále lze nastavit kompresi videa, to lze buď ukládat bez komprese, ale video je potom velmi velké a již velmi krátké video zabírá několik gigabajtů paměti. Další možností tedy je povolit kompresi a nechat na uživateli, ať si vybere, jaká komprese bude použita a úroveň této komprese. To ovšem bohužel není ideální, protože u nastavení některých kompresních metod aplikace přestane fungovat a skončí chybou. U některých se zase přes výsledné video zobrazovaly barevné pruhy. Použitelný tedy byl jenom jeden druh

komprese, u něj ale bylo ještě potřeba najít použitelnou úroveň komprese, aby video bylo co nejmenší a zároveň aby jeho kvalita nebyla moc špatná. Dále bylo potřeba zajistit způsob jak získávat tyto bitmapy pro tvorbu videa. XNA poskytuje možnost vykreslovat místo na obrazovku do textury. Stačí nastavit v Graphics Device cíl pro vykreslování. Tuto texturu je poté potřeba převést na bitmapu. To nakonec nebylo úplně jednoduché. Textury v XNA mají sice metodu pro převedení textury na obraz, ale ta bohužel nefunguje správně. Je v ní chyba, kdy při opětovném použití neuvolňuje paměť a to ani při ručním zavolání garbage collectoru. Aplikace tedy po pár vteřinách běhu přestala fungovat kvůli nedostatku paměti. Bylo tedy potřeba napsat metodu, která převede texturu na bitmapu přímo. Je potřeba přímo překopírovat data z textury do bitmapy. Toto překopírování se provádí v jednoduchém cyklu, který ovšem sníží rychlost algoritmu. Tvorba videa tedy trvá o něco déle.

5. Vytvoření demonstrační scény

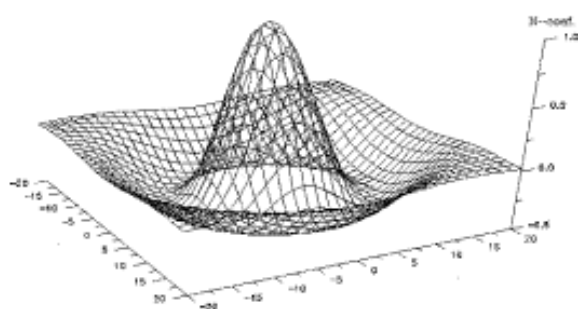
V této kapitole budu postupně popisovat, jak jsem postupoval u tvorby demonstrační scény. To by mělo ukázat, jak se s editorem pracuje a také zároveň se pokusím okrajově popsat některé další jednodušší algoritmy. Snažil jsem se, aby aplikace měla ovládání co nejjednodušší a také aby bylo velmi intuitivní, z toho důvodu je také v horní liště aplikace nápověda. Ta funguje tak, že pokud máme zvolenou nějaký nástroj, objeví se vysvětlující text, jak se daný nástroj používá. Většina tlačítek v editoru má slovní popis co dělají a v případě, že jsou tvořeny ikonou, je použit pro nápovědu tooltip. Ten se objeví, pokud na tlačítko najedeme myši.

Editor umožňuje ukládání a zpětné načítání scén. Soubory se uloží do samostatné složky. Editor při uložení vytváří několik souborů. První je pro uložení výškové mapy, další pro uložení textur a jejich nastavení a poslední pro uložení modelů. Program také uloží samotné textury do zvláštní složky. U modelů se ukládá celá stromová hierarchie a také nastavení daného modelu. Po načtení aplikace se nacházíme v režimu modelování terénu a máme zobrazený základní prázdný terén. Při stisknutí tlačítka nový terén se nám objeví nabídka s volbou velikosti terénu. Při zadání první hodnoty se dopočítá maximální hodnota pro druhou velikost. Pokud bychom chtěli čtvercový terén, můžeme ho mít o největší velikost 2056 x 2056 bodů. Ale je také možné vytvořit pás o velikosti 257 x 16 448 bodů. Dále je možné nastavit pozici radaru a citlivost myši. Pohyb kamerou po terénu lze používat nezávisle na aktuální používané funkci a je možné se tedy pohybovat nad terénem při jakékoliv práci v editoru. Veškerý pohyb, přibližování a otáčení lze provádět pomocí myši, jediným omezením je, že potřebujeme třítláčkovou myš. Abych odlišil totiž pohyb myši pro používání akci editoru, kde jsou využita obě základní tlačítka myši, použil jsem pro pohyby kamery třetí tlačítko. V dnešní době již téměř všechny myši disponují tímto třetím tlačítkem a tím pádem bude uživatel omezen jen v případě použití touchpadu, ten ovšem sám o sobě není moc vhodný k modelování terénu.

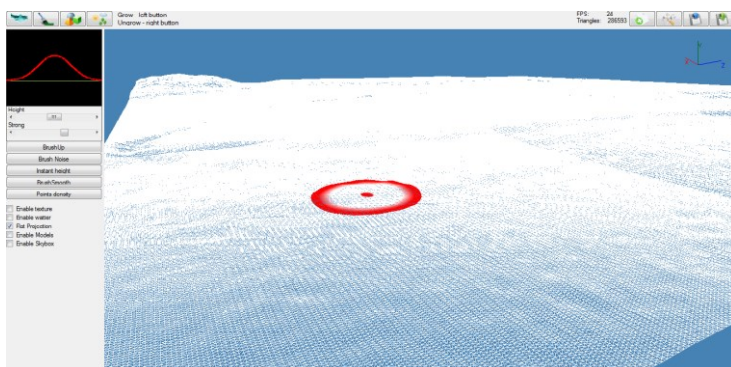
5.1. Modelování terénu

První částí je modelování terénu, to nám určuje samotný tvar, jak bude výsledný terén vypadat. Modelování probíhá použitím jednotlivých štětců. K dispozici máme 5 druhů štětců a u každého se dá ještě nastavit velikost a intenzita štětce. Základní štětec slouží k jednoduché tvorbě kopců, zde se používá funkce mexican hat, který vytváří zaoblené kopce, tvar této funkce je vidět na obrázku 15, další informace nalezneme [9]. Výhodou této funkce je, že lze jednoduše měnit její velikost třemi jednoduchými parametry, které nastavují výšku funkce, šířku a poslední prohloubení na okrajích. Poslední hodnota je pro tvorbu kopců v editoru nevhodná a bylo tedy potřeba najít hodnotu, aby ve funkci tyto prohlubně vůbec nebyly. Druhé dva parametry nastaví intenzitu a velikost štětce. Stejná funkce se používá pro tvorbu prohlubní. Dalším štětcem je „noise“, ten umožňuje vytvářet nerovné části na terénu. Celý princip funguje tak, že na náhodných místech v oblasti, do které zasahuje štětec, se zavolá opět funkce „mexican hat“ s co nejmenším rozsahem velikosti a intenzity. To na

náhodných místech buď terén zvedne, nebo sníží a vytvoří tak několik různých kopců, které působí jako zvlnění terénu. U tohoto štětce můžeme nastavovat jeho rozsah a také určovat pravděpodobnost, s jakou se budou kopce vytvářet. Další štětec umožňuje terén v celém rozsahu štětce nastavit na jednu hodnotu, ta se určí z prvního kliknutí na terén a poté už jenom táhneme myší tam, kde chceme tuto výšku terénu. Výhodou je že můžeme určit podle stisknutého tlačítka myši, jestli chceme terén na tuto hodnotu jenom snižovat, nebo jenom zvyšovat. Další štětec nám umožňuje vyhlazování terénu, ten počítá průměrnou hodnotu výšky ze všech bodů, které štětec pojme při kliknutí do mapy a těm, které mají nižší hodnotu, jejich výšku zvýší a u těch s vyšší hodnotou zase naopak. Tuto hodnotu mění podle vzdálenosti od středu štětce, čím je daný bod blíže ke středu, tím víc se změní jeho hodnota. Poslední štětec dovoluje měnit pozici bodů. A to tak, že můžeme některé body posunout blíž k sobě a vytvořit pak na nějakém místě více detailů, než na jiném. Toto posouvání můžeme použít buď kruhově do středu, nebo podle os horizontálně nebo vertikálně. Nevýhodou této funkce je to že se rozhodí mřížka a je potřeba řešit spoustu problémů. V této části, kde modelujeme terén, máme standardně terén zobrazen jako drátěný model. Ale pod nabídkou štětců máme několik zaškrtávacích políček, ve kterých si můžeme nastavit zobrazování textur, modelů, vody a skyboxu. Pokud si ovšem zapneme textury, nespočítají se v tomto režimu normály. V tomto režimu by se totiž při změně tvaru terénu museli počítat velmi často a to by trvalo dlouho. Ty se tedy počítají vždy, když přepneme z tohoto modelovacího režimu na nějaký jiný. Tento výpočet zabere nějaký čas, kurzor se v té době přepne na čekací kurzor, aby uživatel poznal, že se něco děje. To samé se stane i s objekty, pokud zvýšíme, nebo snížíme terén pod modelem, model zůstane na svém místě a jeho výška se přepočte až ve chvíli, kdy se přepneme do jiného režimu. Ukázkou modelovacího režimu při tvorbě demonstrační scény můžeme vidět na obrázku 16.



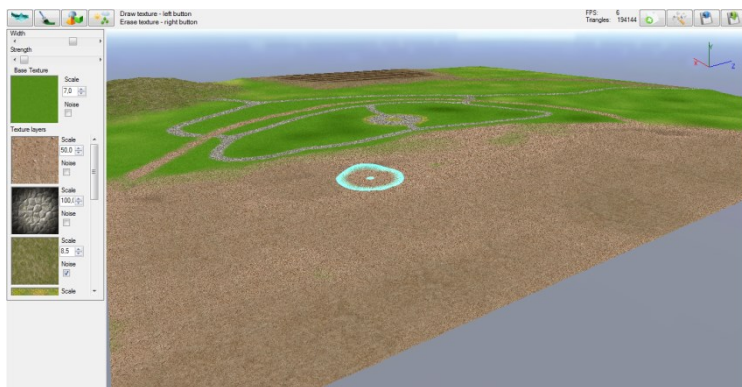
Obrázek 15: Funkce Mexican hat



Obrázek 16: Modelování terénu

5.2. Kreslení textur

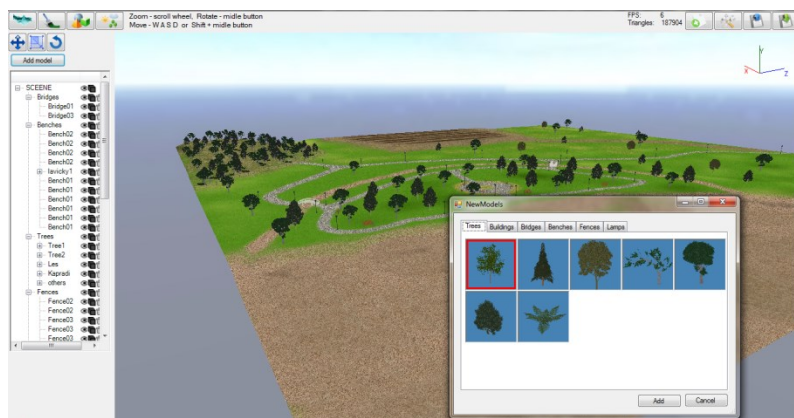
Další část aplikace slouží pro kreslení textur. Při tvorbě nového terénu máme nastavenou akorát základní texturu, která je čtverečková a je ji potřeba nahradit jinou. Textury můžeme načítat z libovolného umístění v počítači. Takových textur můžeme mít celkem 8 a jednu základní, tou kreslit nemůžeme, ale nastaví se automaticky přes celý terén. Ostatní textury můžeme nanášet na terén pomocí štětce, kterému můžeme nastavovat jeho velikost a také úroveň krytí. Pokud nastavíme krytí na menší úroveň, můžeme vytvořit plynulé přechody mezi jednotlivými částmi, kde máme rozdílné textury. Pokud ale nastavíme krytí na maximum, jsme schopni vytvořit ostrý přechod, který se dá například využít při kreslení cest. Výhodou je, že přidané textury vidíme přímo v boxu, ze kterého si vybíráme, kterou texturou budeme kreslit. Každé textuře se dá nastavit opakování a také se dá texturám nastavit jako podklad textura s detailním šumem. Tento šum se zobrazuje podle toho, jak daleko jsme s kamerou vzdáleni od terénu, pokud jsme moc vzdáleni, není vidět vůbec, ale čím víc se přibližujeme, množství šumu se zvětšuje. Při uložení scény editor také uloží všechny použité textury. Ukázka nanášení textur je na obrázku 17.



Obrázek 17: Texturování scény

5.3. Vkládání modelů

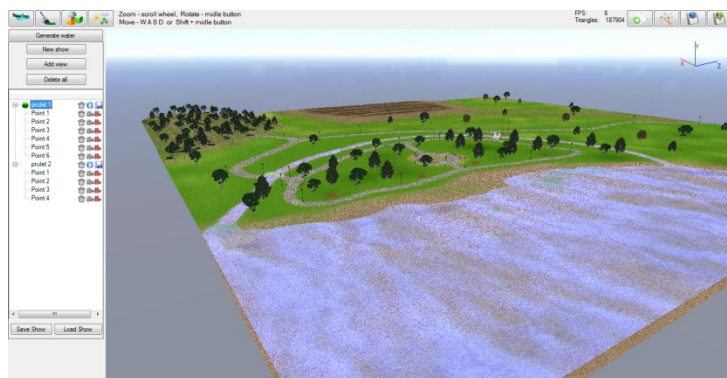
Vkládání modelů se provádí z připravené kolekce. Bohužel XNA nedovoluje načítat externí modely a ty je potřeba mít zkompilovány Visual Studiem a uloženy v Contentu. Tento přístup má také jednu výhodu a to, že tyto zkompilované modely jsou mnohem menší než původní objekty. Nevýhoda je ovšem mnohem větší a to, že se musíme spokojit s těmi modely, které máme přidány do editoru před kompilací. Samotné vkládání modelů probíhá tak, že se nám otevře okno s kolekcí modelů, v něm jsou v několika složkách podle typu zobrazeny náhledy modelů, to je vidět na obrázku 18. Po vybrání modelu vkládáme tento model do scény klikáním myši. Vybraný model můžeme vložit vícekrát na různá místa, dokud vkládání nezrušíme. Každý vložený model, si nastaví hodnotu výškové souřadnice podle úrovně terénu, na které jsme ho vložili. Modely lze vkládat do scény tak, že se modely budou sami vkládat do složky rozděleny podle svého typu. Druhou možností je, vkládat tyto modely přímo do složek, které si vytvoříme sami, nebo již vytvořeny jsou. Po dobu vkládání jednoho druhu modelu se budou modely vkládat do stejné složky. Tyto modely je možné po vložení také dále přesunovat do jiných složek. Dále je možné každému modelu, nebo i všem modelům v jedné složce měnit pozici na terénu, otočení modelu, velikost modelu a také hodnotu výškové souřadnice. Dále můžeme pomocí vykreslovacího stromu nastavit skrytí modelů, nebo jejich zobrazení v režimu drátěného modelu. Toto můžeme nastavovat pro jednotlivé modely, ale také i pro celou složku. V tomto stromě nalezneme i ikonku pro smazání modelů.



Obrázek 18: Vkládání modelů do scény

5.4. Vložení vody a vytvoření průletu scénou

Poslední část editoru slouží k vkládání vody a k tvorbě průletu scénou. Při vkládání vody se nejprve zobrazí obdélník s vodní hladinou přes celý terén. Výšku této vodní hladiny pak upravujeme pomocí šipek, na terénu jsou vidět místa, na kterých bude voda nad terénem. Uživatel pak kliknutím určí, kde voda doopravdy bude, respektive může vodní hladinu pravým tlačítkem myši také vymazat. Pokud uživatel vloží vodu do místa, kde už voda je, a to třeba s jinou výškou hladiny, ta se prve smaže a vloží se voda nová. V této části se také vytvářejí průlety scénou. Současně je možné mít vytvořeno několik různých průletů scénou. Tyto průlety se vytvářejí vkládáním jednotlivých bodů průletu. Ty určujeme tak, že se kamerou pohneme do bodu, který chceme přidat a vložíme tento bod do aktivního průletu. Ten se pak zařadí na konec seznamu bodů. Postupně se interpolací vytváří průlet mezi jednotlivými body. Doba průletu mezi jednotlivými body je spočtena podle posunu, rotace a oddálení kamery mezi jednotlivými body. Samotný průlet je možné si prohlédnout kliknutím na ikonku spuštění u daného průletu. Pokud se při tvorbě průletu něco nepovedlo je možné body posouvat na jiné pozice a to i dokonce mezi jednotlivými průlety, mazat nebo i měnit nastavení kamery v jednotlivých bodech. Pokud máme průlet hotový a chceme ho uložit, je možné ho uložit samostatně, kdy se ukládají jednotlivé body do souboru, ze kterého můžeme tento průlet kdykoliv znovu načíst nezávisle na terénu, který máme otevřený. Druhou možností je vytvořit z tohoto průletu prezentační video. Toto video se uloží na disk a je ho možné spouštět již nezávisle na editoru v přehrávači videí.



Obrázek 19: Vkládání vody a vytvoření průletů scénou

6. Výkonnostní testy a jejich vyhodnocení

Dále jsem provedl testování aplikace. Prováděl jsem dva druhy testů, první sleduje počet fps a druhý testuje využití paměti RAM. Tyto testy proběhly na notebooku, jehož parametry nalezneme v tabulce 2. Před testováním jsem na počítači vypnul všechny další aplikace, které by mohli značně zatěžovat počítač a zakreslit tak hodnoty testu. Velikost využití paměti RAM jsem odečítal ze správce úloh ve Windows, tyto hodnoty tedy nemusí být úplně přesné a jsou spíše orientační. Tyto testy by také měli pomoci nalézt minimální požadavky pro konfiguraci počítače.

Notebook	HP ProBook 4525s
Procesor	AMD Turion(tm) II P560 Dual-Core
Takt procesoru	2,50 GHz
RAM	4 GB DDR3
Grafická karta	ATI Mobility Radeon HD 5470
Velikost vlastní paměti grafické karty	512 MB
Operační systém	Windows 7 Professional 64bit

Tabulka 2: Testovací notebook

6.1. Popis testování

Testování proběhlo v několika fázích, v každé z těchto fází jsem sledoval využití paměti RAM počet FPS a v některých testech i také počet vykreslených trojúhelníků. Pro počet FPS zaznamenávám dvě hodnoty a to z důvodů, že tato hodnota často kolísá. Nechal jsem tedy vždy aplikaci chvíli běžet a sledoval jsem minimální a maximální počet FPS.

První test zjišťoval závislost výkonu na velikosti terénu, ten jsem vykresloval v režimu drátěného modelu a postupně jsem měnil jeho velikost, mezi každým testem jsem aplikaci vypnul a pak znovu spustil, aby nezůstala nějaká neuvolněná paměť. Pro tento test jsem volil čtvercové velikosti terénu. Ostatní testy probíhaly vždy na terénu o velikosti 512 x 512 a terén byl zobrazen v režimu textur. Ten sám o sobě má nižší počet FPS než režim drátěného modelu, tento rozdíl v testech také zohledňuji a také tento rozdíl popíšu. Dále následuje test využití textur, u něj očekávám, že by počet použitých textur neměl mít žádný nebo aspoň jenom minimální vliv na výkon aplikace. Textury budu přidávat do editoru postupně a každou texturou obarvím jednu část terénu. V tomto testu budu také zkoumat, jestli má na výkon vliv použití textury detailů. Další test bude zjišťovat, jak se projeví počet modelů na výkonu aplikace. Zde také ukážu, že velmi závisí na každém modelu a na kvalitě jeho zpracování.

Pak přijde na řadu testování použití vody, zde budu testovat závislost výkonu na velikosti vodní hladiny. Poslední test bude trochu rozdílný od ostatních testů, bude totiž testovat tvorbu prezentačního videa. Zde budu prvé sledovat velikost výsledného videa a délku jeho tvorby v závislosti na jeho délce. A pak také budu zjišťovat vliv komprese na velikosti videa a také na jeho kvalitu.

6.2. Test velikosti terénu

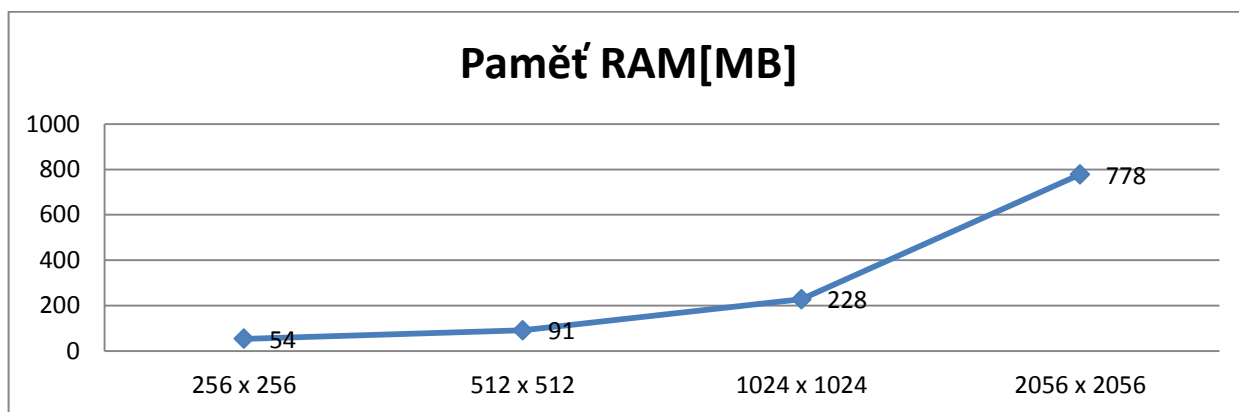
V tabulkách 3 a 4 jsou výsledky testů, kde jsem měnil velikost terénu a sledoval jsem, jak se mění počet FPS a počet trojúhelníků, který je vykreslován. V první tabulce jsem měl kameru nastavenou nad předním rohem terénu a v záběru byl celý terén. V druhé tabulce jsem nastavil kameru nad středem terénu tak aby byl celý vidět. Je vidět, že díky oddálení k tomu abych shora viděl celý terén, způsobilo LOD mnohem lepší výsledky, než v první tabulce. Dále je na grafu 1 vidět vytížení paměti RAM v závislosti na velikosti terénu. Hodnoty u menších terénů neobsahují pouze velikost obsahující samotný terén, ale také velikost, kterou zabírá samotné XNA. Dále je vidět, že při maximální velikosti terénu, využívá již 778 MB paměti, to určuje minimální požadavky na paměť, musíme samozřejmě ještě přičíst další paměť pro modely, vodu a textury.

Velikost terénu	Min FPS	Max FPS	Počet trojúhelníků
256 x 256	60	65	33149
512 x 512	41	42	132596
1024 x 1024	32	35	187904
2056 x 2056	30	32	214640

Tabulka 3: Test velikosti terénu v pohledu z rohu na celý terén

Velikost terénu	Min FPS	Max FPS	Počet trojúhelníků
256 x 256	70	74	33149
512 x 512	68	72	33524
1024 x 1024	80	84	8912
2056 x 2056	65	68	35648

Tabulka 4: Test velikosti terénu v pohledu nad terénem



Graf 1: Paměť RAM v závislosti na velikosti terénu

6.3. Test textur

V tabulce 5 jsou výsledky z testu použití textur. V tomto případě jsem měl kameru opět nad terénem a vidět byl celý terén. Postupně jsem na terén nanášel textury tak, aby zabírali stejnou část terénu. Ten to test už podle očekávání ukázal, že počet textur má na počet FPS i velikost využité paměti RAM minimální vliv. Jediné co můžeme poznat je velký rozdíl mezi zobrazením v režimu drátěného modelu, nebo samotné textury. To můžeme porovnat s tabulkou 4, ve které byl použit stejný pohled kamery. Zde porovnáme velikost terénu s hodnotou 512 x 512, kde jsou FPS o skoro 30 vyšší. Dále jsem zkoušel, jestli bude mít vliv zobrazení textury detailů, ale test neodhalil vůbec žádný rozdíl.

Počet textur	Min FPS	Max FPS	RAM [MB]
S přednastavenou základní texturou	45	47	80
Vlastní základní textura	45	47	91
1 další textura	45	46	105
2 další textury	45	46	102
3 další textury	43	44	118
4 další textury	43	44	116
5 dalších textur	44	45	113
6 dalších textur	43	44	114
7 dalších textur	44	45	117
8 dalších textur	44	45	121

Tabulka 5: Test použití textur

6.4. Testování modelů

Další dva testy se týkaly modelů, zde se ukázalo, že je velmi důležité kolik má daný model polygonů a jak moc zpomalují celý editor modely s hodně polygony. Tyto výsledky jsou v tabulkách 6 a 7. V první tabulce jsem přidával do scény stromy, a je vidět, že už i při 30 objektech klesne počet FPS na polovinu, kdežto v druhé tabulce kdy jsem do scény přidával pouze částí plotu, tak se počet FPS změnil minimálně. Dále je v tabulkách vidět, že počet přidávaných modelů nemá vůbec žádný vliv na velikost využití paměti RAM. To je dáno tím, že se každý model načítá jenom jednou a dále se už jenom používají reference na tento načtený model.

Počet modelů	Min FPS	Max FPS	RAM [MB]
0	46	48	80
1	44	46	83
10	37	38	89
30	25	27	84
50	20	20	86

Tabulka 6: Test modelů s hodně polygony

Počet modelů	Min FPS	Max FPS	RAM [MB]
0	45	46	86
1	42	46	86
10	44	45	86
30	44	45	86
50	42	44	86

Tabulka 7: Test modelů s málo polygony

6.5. Test vody

V tabulce 8 je vidět nejslabší část tohoto editoru a tou je vykreslování vody. Její vkládání sice funguje lépe, než v ostatních editorech, ale to si také vybírá svou daň. Opět jsem měl k dispozici terén o velikosti 512 x 512 ten jsem postupně zaplavoval vodou. Nevýhodou vody je, že na ní není použito žádné LOD a vykresluje tedy pořád všechny trojúhelníky. Navíc je voda tvořena ze stejných trojúhelníků jako terén, takže i když je tato voda zobrazena přes celý terén a stačily by tyto

trojúhelníky jenom dva, je jich použito mnohonásobně víc. To všechno se také odráží na výsledném snížení výkonu a spolu s přidáním mnoha stromů dokáže snížit FPS i klidně na 10.

Zaplňená oblast vodou	Min FPS	Max FPS	RAM [MB]	Celkový počet trojúhelníků ve scéně
0	45	46	84	33 524
1/4	34	36	109	169 156
2/4	29	30	133	298 530
3/4	23	25	142	427 535
4/4	21	22	174	557 812

Tabulka 8: Test vody

6.6. Testování prezentačního videa

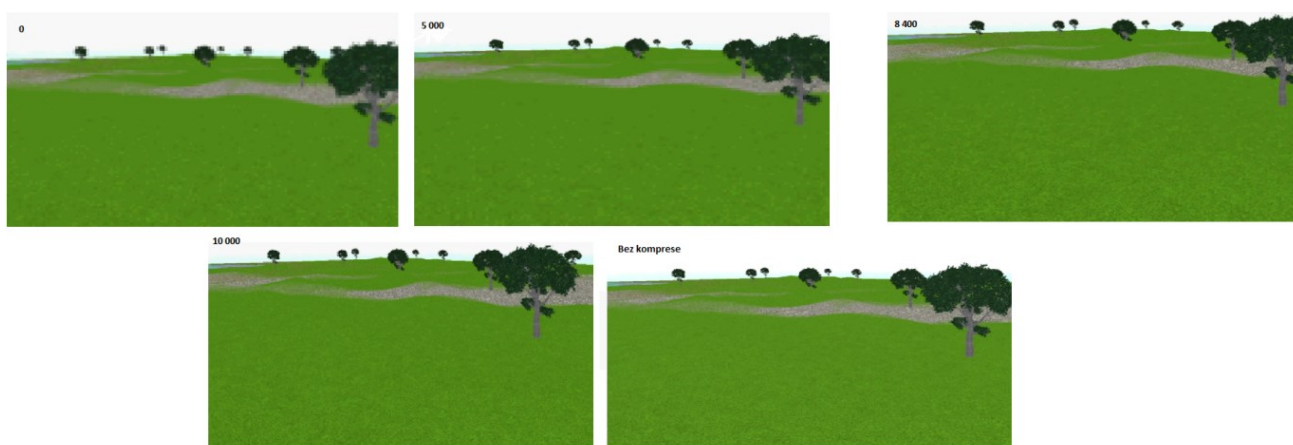
V posledním testu jsem testoval tvorbu videa. První tabulka 9 ukazuje velikost videa a dobu trvání tvorby tohoto videa. Video bylo tvořeno opět na terénu o velikosti 512 x 512 s namodelovaným terénem, několika objekty a vodou. Tabulka 10 zobrazuje velikost videa podle nastavení komprese. Kde první možnost je bez komprese a další jsou nastavovány v rozmezí 0 až 10 000, kde 0 je největší komprese a 10 000 nejmenší. Dále je na obrázku 20 vidět porovnání kvality videa podle nastavení komprese. Z toho lze vidět, že sice při maximální kompresi je video mnohem menší, ale video je velmi nekvalitní. Musel jsem tedy testováním nalézt ideální nastavení komprese, aby video mělo co nejmenší velikost a zároveň kvalita tohoto videa byla dostačující. Nakonec jsem použil hodnotu 8 400, kdy u této hodnoty komprese má ještě video dostatečně malou velikost a zároveň se komprese neprojeví tolik na kvalitě.

Délka videa [mm:ss]	Doba tvorby videa [mm:ss]	Velikost videa [MB]
00:04	00:15	9,15
00:14	00:48	40,9
00:24	01:12	62,5
00:51	02:34	129

Tabulka 9: Velikost videa a délka tvorby v závislosti na délce videa

Nastavení komprese	Velikost videa [MB]
Bez komprese	650
10 000	130
8 400	74,8
5 000	21,6
0	20,3

Tabulka 10: Nastavení komprese



Obrázek 20: Kvalita videa podle komprese

6.7. Vyhodnocení testů

Předchozí testy ukázali, že editor může běžet i na průměrném stroji. Ale také nám určil některé minimální požadavky. První důležitý požadavek je na paměť RAM, zde jsem se editorem dlouhodobě nedostal nad 1GB využití paměti, ovšem krátkodobě může editor využívat až 1,5 GB paměti, to může nastat, pokud vytvoříme maximálně velký terén rychle dvakrát po sobě, nedá se sice očekávat, že by to při normálním používání stalo, ovšem musíme počítat s tím, že to může nastat. Dalším požadavky určuje samotné XNA, to se dá používat ve dvou režimech Reach a HiDef. Reach slouží pro jednodušší aplikace a má také menší požadavky na grafickou kartu. Já jsem pro editor použil režim HiDef, ten poskytuje více možností, umožňuje použít větší textury pro vykreslování. Dále umožňuje najednou vykreslit více trojúhelníků, kde v profilu Reach je možné vykreslit jenom 65 535 trojúhelníků a v režimu HiDef až 1 048 575 trojúhelníků, a také dále několik dalších rozšíření. To ale také určuje minimální požadavky na grafickou kartu, ta musí podporovat DirectX 10 a na internetových stránkách Microsoftu určují jako minimální požadavky na tuto kartu, aby

poskytovala minimálně stejné možnosti jako grafická karta v Xbox 360. Další požadavky jsou na velikost volného místa na disku, samotná aplikace zabírá na disku pouhých 67 MB, přesto, že aplikace obsahuje již vloženou kolekci modelů. Visual Studio totiž při kompilaci také zkomprimuje velikost celého Contentu, který má původně velikost 153 MB a po kompresi 66 MB. Ten obsahuje kromě modelů také i efekty a základní textury. Z toho také vyplývá, že Content zabírá největší část celé aplikace a ostatní soubory mají dohromady asi jenom 1 MB. Další požadavky na místo na disku je potřeba pro ukládání terénu a také vytvořených videí. Velikost videa je často mnohem větší, než je velikost celé aplikace.

7. Shrnutí dosažených výsledků

V kapitole [Cíle](#) jsem si vytkl vytvořit 3D editor terénu s jednoduchým ovládáním. A mohu říci, že to se v mnohém povedlo. Existuje sice ještě několik úprav, které by se na tomto editoru dali udělat. Ale většina funkcí, které byly zadané, se podařilo v editoru implementovat. Je možné vytvářet různě velké terény. Terén lze efektivně modelovat pomocí několika štětců. Lze na něj nanášet až 8 textur plus jednu základní. Můžeme vkládat modely, to je bohužel omezeno na modely, které jsou přednastaveny v editoru, ale to je dáno omezením samotného XNA. Vytvořil jsem také jednoduchou správu scény, pro ovládání těchto modelů. Editor dále umožňuje vkládat do scény vodu na vybraná místa s různou výškou vodní hladiny. Výstupem z tohoto editoru je možnost uložení si scény pro opětovné načtení, nebo vytvoření prezentačního videa. Jedinou slabinou tohoto editoru, je jeho výkonová náročnost. Editor sice spustíme na většině novějších počítačů. Ovšem jeho výkonost klesá s množstvím použitých modelů a velikostí vody. Náročnost modelů je dána tím, že tyto se jedná o bezplatné modely stažené z webových kolekcí a tedy jejich kvalita není moc vysoká. Pro editor jsem také vytvořil instalační balíček. Ten na instalovaném stroji zkontroluje, zda je nainstalován .Net Framework a XNA. Pokud ne navrhne uživateli stáhnutí těchto komponent z internetu a sám je nainstaluje.

8. Další vývoj editoru

Zde popíšu, jakým směrem by se měl udávat další vývoj editoru. Popíšu části, které by se mohli vylepšit, nebo se dá pokračovat v jejich vývoji. Editor splňuje sice všechny body, které byly dány v zadání, ale stále některé části mohou být zpracovány tak, aby aplikace fungovala lépe, nebo je možné jinak rozšířit jejich funkčnost nad úroveň zadání.

8.1. Import a export terénu

V části importování, by se mohla přidat možnost načítání externí výškové mapy. To umožní načítat mapy z externích dat, které jsme získaly například z jiných aplikací. Tyto výškové mapy bývají vytvořeny ve formě černobílého obrázku. Tento obrázek má velikost podle výsledné velikosti terénu, a každý bod obsahuje hodnotu podle jeho výšky. S této mapy se pak dá jednoduše načíst 3D terén. Bude ovšem potřeba řešit některé problémy, například mapa v editoru se skládá z jednotlivých čtverců a ty mají pevnou velikost, jak tedy načíst terén, jehož velikost nebude násobkem použitých čtverců.

U exportování dat z editoru, můžeme přidat zase ukládání do výškové mapy. Což bude obrácený postup než načítání, ale zde není potřeba řešit žádné omezení jako u načítání. Dále by aplikace mohla výsledný terén exportovat jako objekt v různých formátech. To samozřejmě vyžaduje vytvoření modulu, který bude obstarávat tento export do jednotlivých formátů. To by umožnilo větší využití aplikace, protože tyto modely by se mohly dále používat v dalších aplikacích, které umožňují import těchto objektů.

8.2. Vykreslování terénu

První problém u vykreslování terénu je jeho maximální velikost. Ta je dána tím, že všechny vertexy jsou drženy v poli a pokud bychom vytvořili větší terén, tak by již počet vertexů přerostl maximální velikost pole. Existuje několik způsobů jak vykreslovat tyto větší terény. Jednou jednoduchou možností je vytvořit si těchto polí několik a rozdělit tento terén na části a každou držet ve vlastním poli. To ovšem zase způsobuje několik problémů, kdy první je upravování samotného terénu, pokud bychom ho upravovali na okraji tohoto pole, bude potřeba řešit přechody mezi jednotlivými poli. Další problém bude paměťová náročnost, kdy stávající největší terén využije skoro 1 GB paměti. To tedy zvýší také základní požadavky na editor. Další možností je rozdělit si terén na části a ty načítat postupně tak jako v editoru PNP terrain creator. To sice nezvýší náročnost aplikace, ale je potřeba zase přepracovat celý systém přepínání mezi těmito částmi. A také je potřeba řešit navazování terénu mezi těmito částmi. Respektive u průletu scénou je potřeba řešit navazování terénu v reálném čase.

Dalším problémem u vykreslování terénu je zpracování LOD. To je nyní zpracováno metodou Geo Mip-Mapping, ta je sice efektivní a velmi jednoduchá, ale nepatří mezi nejlepší a také tím, že vždy celý čtverec má stejnou úroveň zobrazení, jsou vidět skokové rozdíly a to nevypadá moc dobře. Výhodné by tedy bylo zpracovat některý jinou metodu pro LOD, z dříve popisovaných by byla vhodná metoda ROAM, ta se zpracovává v reálném čase a její výhodou je, že je spojitá a nevznikají tedy díry v terénu a není potřeba je vyplňovat.

8.3. Kreslení textur

Kreslení textur má omezení v maximálním počtu použitých textur. Toto vkládání je možno zvětšit přístupem, kdy všechny textury kreslíme do jedné výsledné textury, tento přístup umožňuje vykreslovat neomezený počet textur. Problém je, že nakreslené textury nemůžeme mazat, ale musíme je překreslit jinou texturou. Dalším vylepšením by mohla být možnost volit si druh použité detailní textury, nyní je k dispozici jen jedna přednastavená.

8.4. Použití objektů

Největší problém u vkládání modelů je omezení počtu modelů jen na již vytvořenou kolekci. Při tvorbě reálného terénu potřebuje uživatel často vkládat další externí modely. Další úpravou by tedy mělo být zpracování tohoto problému. Pro tento problém již existuje několik řešení, jak obejít tohle omezení, tenhle přístup samozřejmě funguje pouze na Microsoft Windows, což pro editor naprosto dostačuje, způsob řešení tohoto problému nalezneme v [10] a [11]. Dalším vylepšením by mohla být změna ovládání modelů. Ty se nyní ovládají po označení pomocí tlačítek klávesnice. Lepší přístup by byl pomocí ovládání myši, kdy se zobrazí u modelu šipky a pak myši můžeme pomocí těchto šipek měnit pozici, velikost, nebo rotace modelu. Posledním vylepšením u modelů, by mohl být generátor náhodného rozmístění modelů, kde by si uživatel vybral oblast a počet přidávaných modelů a tento generátor by tyto modely náhodně rozmístil. To by sloužilo například k jednoduché tvorbě hustého porostu.

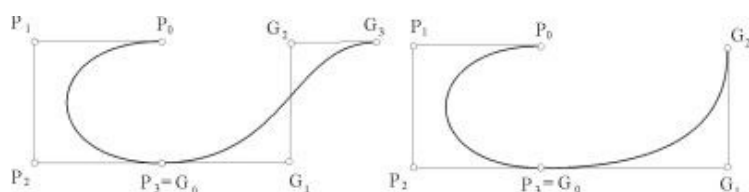
8.5. Voda

Výhody a nevýhody, které má zobrazování vody v editoru jsem již popsal v předchozích kapitolách. Pro rychlejší běh aplikace by tedy bylo potřeba implementovat vodu jiným způsobem, než je tomu teď. U vody totiž není potřeba vkládat tolik trojúhelníků, kolik se vkládá nyní. Problém je, že u semínkového vyplňování se špatně řeší spojování trojúhelníků. Jedním řešením by bylo použít jinou verzi tohoto vyplňování, která vyplňuje ne po bodech, ale po celých pásech. Tento pás bych pak

vyplnil dvěma trojúhelníky, to by mnohem zmenšilo počet výsledných trojúhelníků. Problém tohoto přístupu nastane, pokud použijeme štětec, který posunuje vertexy, ten totiž rozhodí pravidelnou mřížku, kterou je pro tento přístup vyplňování potřeba. A u nepravidelné mřížky, která tak vznikne, se bohužel nedá použít. Bylo by tedy potřeba nalézt jiný algoritmus, který by zmenšil počet vykreslovaných trojúhelníků. Dále by se dalo pro vodu napsat vlastní shader, který by přidal vodě realistický vzhled.

8.6. Průlety scénou

V průletech scénou se dnes vytváří samotný průlet mezi body pomocí lineární interpolace mezi jednotlivými body, to ovšem způsobuje, že změny směru po každém bodu působí trhaně. Řešením by bylo použít pro interpolaci mezi body Bézierovy křivky, ty umožňují vytvořit průlety mezi jednotlivými body spojitě bez trhání. Průlety kamerou po křivce totiž vypadají více přirozeně. Bézierovy křivky můžeme vidět na obrázku 21.



Obrázek 21: Bézierovy křivky

9. Závěr

V diplomové práci na téma Editor terénu s použitím XNA, jsem se zabýval zkoumáním současných 3D editorů terénu. Porovnáním těchto editorů a návrhem a implementací vlastního editoru. Tento editor jsem dále otestoval a vytvořil v něm ukázkovou scénu, pro prezentaci možností editoru.

V kapitole [Další vývoj editoru](#) jsem dále navrhnul, několik dalších úprav, kam by se mohl ubírat další vývoj editoru.

Tento editor je použitelný pro tvorbu prezentačních videí, v oblasti tvorby ukázek parků, golfových hřišť a dalších podobných oblastí. A to buď jako prezentaci terénu pro zákazníky, před reálnou tvorbou těchto terénů, nebo jako ukázky existujících terénů. Editor poskytuje jednoduchý přístup k ovládání aplikace bez nutnosti složitého učení se ovládat aplikaci. A také obsahuje většinu možností, které poskytuje většina současných editorů na trhu.

10. Seznam zkratek a pojmů

2D/3D	způsob zobrazení objektů, dvourozměrně nebo třírozměrně
Buffer	vyrovnávací paměť, slouží většinou k dočasnému uchovávání dat, před jejich přesunem na jiné místo
FPS	frames per second, počet snímků za sekundu, určuje plynulost běhu grafických aplikací
Framework	softwarová struktura, která slouží jako podpora při programování, může obsahovat podpůrné programy, API, podporu pro návrhové vzory, a podobně
Garbage collector	využívají ho vyšší programovací jazyky, slouží k automatickému mazání již nepoužívané paměti
LOD	level of detail, nastavení úrovně detailů, zmenšuje počet trojúhelníků potřebných k vykreslení u vzdálených objektů, nebo částí terénu
MB	megabyte, jednotka dat
RAW	surová data
Skybox	v počítačové grafice při zobrazování terénu se používá k zobrazení vzdáleného okolí, vytvoří se velká krychle, která pojme celou scénu a na její vnitřní stěny se zobrazí textura okolí
Shader	program, který slouží k řízení jednotlivých částí programovatelného vykreslovacího řetězce grafické karty
Trial verze	bezplatná verze jinak placeného programu, která ale obsahuje omezení, neobsahuje všechny funkce nebo jsou omezené
TreeView	složena komponenta knihovny Windows Forms frameworku .Net, ta slouží k textovému zobrazení stromové struktury
Vertex	základní zobrazovací jednotka v prostoru, označuje jeden bod

11. Seznam obrázků

OBRÁZEK 1: TERRAEd EDITACE TERÉNU	4
OBRÁZEK 2: TERRAEd ZOBRAZENÍ DRÁTĚNÉHO MODELU.....	4
OBRÁZEK 3: TERRAEd SPRÁVA SCÉNY.....	5
OBRÁZEK 4: UKÁZKA EDITORU EARTH SCULPTOR	6
OBRÁZEK 5: PNP 2D ZOBRAZENÍ TERÉNU	8
OBRÁZEK 6: PNP 3D ZOBRAZENÍ UKÁZKOVÉ SCÉNY.....	8
OBRÁZEK 7: T.ED	9
OBRÁZEK 8: GROME OVLÁDACÍ PROSTŘEDÍ	10
OBRÁZEK 9: TERRAGEN 2	10
OBRÁZEK 10: TRIANGLE STRIP	13
OBRÁZEK 11: 3 ÚROVNĚ GEO MIP-MAPPING LOD.....	16
OBRÁZEK 12: SEMÍNKOVÉ VYPLŇOVÁNÍ	16
OBRÁZEK 13: LOGICKÁ STRUKTURA MODELŮ TŘÍDNÍ DIAGRAM.....	19
OBRÁZEK 14: ADVANCED TREEVIEW	20
OBRÁZEK 15: FUNKCE MEXICAN HAT.....	23
OBRÁZEK 16: MODELOVÁNÍ TERÉNU	24
OBRÁZEK 17: TEXTUROVÁNÍ SCÉNY	24
OBRÁZEK 18: VKLÁDÁNÍ MODELŮ DO SCÉNY.....	25
OBRÁZEK 19: VKLÁDÁNÍ VODY A VYTVÁŘENÍ PRŮLETŮ SCÉNOU	26
OBRÁZEK 20: KVALITA VIDEO PODLE KOMPRESCE	32
OBRÁZEK 21: BÉZIEROVY KŘIVKY.....	37

12. Seznam tabulek, grafů a zdrojových kódů

TABULKA 1: POROVNÁNÍ VERZÍ EDITORU EARTH SCULPTOR	5
TABULKA 2: TESTOVACÍ NOTEBOOK	27
TABULKA 3: TEST VELIKOSTI TERÉNU V POHLEDU Z ROHU NA CELÝ TERÉN	28
TABULKA 4: TEST VELIKOSTI TERÉNU V POHLEDU NAD TERÉNEM	28
TABULKA 5: TEST POUŽITÍ TEXTUR	29
TABULKA 6: TEST MODELŮ S HODNĚ POLYGONY	30
TABULKA 7: TEST MODELŮ S MÁLO POLYGONY.....	30
TABULKA 8: TEST VODY	31
TABULKA 9: VELIKOST VIDEO A DÉLKA TVORBY V ZÁVISLOSTI NA DÉLCE VIDEO	31
TABULKA 10: NASTAVENÍ KOMPRESCE	32
GRAF 1: PAMĚŤ RAM V ZÁVISLOSTI NA VELIKOSTI TERÉNU	29
ZDROJOVÝ KÓD 1: SEMÍNKOVÉ VYPLŇOVÁNÍ.....	17

13.Literatura

- [1] Inaki Ayucar. *Easy Rendering with XNA Inside a Windows Form*. 16. 11. 2007, <<http://www.codeproject.com/Articles/21330/Easy-Rendering-with-XNA-Inside-a-Windows-Form>>.
- [2] Cantón Ferrero. *XNA and Windows Forms*. 06. 09. 2008, <[http://xnacommunity.codeplex.com/wikipage?title=XNAEditorEnglish&referringTitle=XNA Editor](http://xnacommunity.codeplex.com/wikipage?title=XNAEditorEnglish&referringTitle=XNA+Editor)>
- [3] *Triangle Strips*. 3. 6. 2012, <[http://msdn.microsoft.com/en-us/library/windows/desktop/bb206274\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb206274(v=vs.85).aspx)>
- [4] Shawn Hargreaves. *Reach vs. HiDef*. 12.3. 2010, <<http://blogs.msdn.com/b/shawnhar/archive/2010/03/12/reach-vs-hidef.aspx>>
- [5] Radek Bartoň. *Modern Algorithms for Real-Time Terrain Visualization on Commodity Hardware*. 2010, < http://geoinformatics.fsv.cvut.cz/gwiki/Modern_Algorithms_for_Real-Time_Terrain_Visualization_on_Commodity_Hardware>
- [6] Studenti Univerzity Palackého v Olomouci. *Geometrie/Vyplňování*. 11. 11. 2009, <<http://cs.wikibooks.org/wiki/Geometrie/Vyplňování>>
- [7] Andrey Gliznetsov. *Advanced TreeView for .NET*. 10. 6. 2006, <<http://www.codeproject.com/Articles/14741/Advanced-TreeView-for-NET>>
- [8] Corinna John. *A Simple C# Wrapper for the AviFile Library*. 4. 1. 2006, <<http://www.codeproject.com/Articles/7388/A-Simple-C-Wrapper-for-the-AviFile-Library>>
- [9] *OpenGL Programming/Scientific OpenGL Tutorial*. 04. 22. 3. 2012, http://en.wikibooks.org/wiki/OpenGL_Programming/Scientific_OpenGL_Tutorial_04
- [10] *XNA Content Tracker*. 6. 3. 2009, <http://contenttracker.codeplex.com/SourceControl/changeset/view/20704#346981>
- [11] *WinForms Series 2: Content Loading*. 1. 10. 2008, http://create.msdn.com/en-US/education/catalog/sample/winforms_series_2